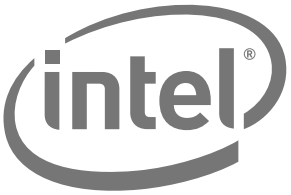

Intel® Pentium® Silver and Intel® Celeron® Processors

Datasheet - Volume 2 of 2

For Volume 1 of 2, refer Document: 336560

Revision 001

February 2018

Document Number: 336561-001

You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

No computer system can be absolutely secure. Intel does not assume any liability for lost or stolen data or systems or any damages resulting from such losses.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

Copies of documents which have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or visit www.intel.com/design/literature.htm.

Intel, Intel® Pentium®, Intel® Celeron®and the Intel logo are trademarks of Intel Corporation in the U.S. and/or other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2018, Intel Corporation. All Rights Reserved.

Contents

1 Introduction 142

2 Host Bridge/DRAM Registers **143**

2.1 Device ID and Vendor ID Register (DEVICE_ID_VENDOR_ID_0_0_0_PCI) — Offset 0h ... 143
2.2 PCI Status and PCI Command Register (PCI_STATUS_COMMAND_0_0_0_PCI) — Offset 4h 143
2.3 PCI Revision ID and PCI Class Code Register (REVISION_ID_CLASS_CODE_0_0_0_PCI) — Offset 8h 144
2.4 Master Latency Timer and Header Type Register (MASTER_LATENCY_TIME_0_0_0_PCI) — Offset Ch 144
2.5 System Agent Memory Mapped Range Base Address Register (SABAR) — Offset 10h 145
2.6 PCI Subsystem Vendor ID and PCI Subsystem ID (SVID_SID_0_0_0_PCI) — Offset 2Ch. 145
2.7 Capability Register Pointer (CAPPTR_0_0_0_PCI) — Offset 34h 145
2.8 B-Unit Copy of the MCHBAR (B_CR_MCHBAR_LO_0_0_0_PCI) — Offset 48h..... 146
2.9 B-Unit Copy of the MCHBAR (B_CR_MCHBAR_HI_0_0_0_PCI) — Offset 4Ch 146
2.10 B-Unit Shadow of GGC (B_CR_GGC_0_0_0_PCI) — Offset 50h 147
2.11 B-Unit Shadow of the DEVEN Register (B_CR_DEVEN_0_0_0_PCI) — Offset 54h..... 148
2.12 Protected Audio Video Path Control (PAVPC_0_0_0_PCI) — Offset 58h 148
2.13 B-Unit PCI Express Enhanced Configuration Range Base Address Low (B_CR_PCIEX-BAR_LO_0_0_0_PCI) — Offset 60h 150
2.14 B-Unit PCI Express Enhanced Configuration Range Base Address High (B_CR_PCIEX-BAR_HI_0_0_0_PCI) — Offset 64h 150
2.15 Message Signaled Interrupts Capability Identifier and Message Control (MSI_CAPID) — Offset 90h 151
2.16 Message Address (MA) — Offset 94h 151
2.17 Message Data (MD) — Offset 98h 151
2.18 Error Status Register (ERRSTS) — Offset A0h..... 152
2.19 Error Command Register (ERRCMD) — Offset A4h 152
2.20 Top of Upper Usable DRAM Low (B_CR_TOUUD_LO_0_0_0_PCI) — Offset A8h 152
2.21 Top of Upper Usable DRAM High (B_CR_TOUUD_HI_0_0_0_PCI) — Offset ACh..... 153
2.22 Base of Data Stolen Memory (BDSM_0_0_0_PCI) — Offset B0h 153
2.23 Base of Graphics Stolen Memory (B_CR_BGSM_0_0_0_PCI) — Offset B4h 154
2.24 B-Unit Copy of the TSEG Memory Base (B_CR_TSEGMB_0_0_0_PCI) — Offset B8h .. 154
2.25 Top of Lower Usable DRAM (B_CR_TOLUD_0_0_0_PCI) — Offset BCh 155
2.26 B-Unit Copy of the MCHBAR (B_CR_MCHBAR_LO_SHADOW_0_0_0_PCI) — Offset D0h... 155
2.27 B-Unit Copy of the MCHBAR (B_CR_MCHBAR_HI_SHADOW_0_0_0_PCI) — Offset D4h ... 156
2.28 Message Data Register (MCRX) — Offset D8h 156
2.29 Scratchpad (SKPD_0_0_0_PCI) — Offset DCh 157
2.30 Capability ID0 Capability Control (CAPID0_CAPCTRL0_0_0_0_PCI) — Offset E0h 157
2.31 Capability ID0 A (B_CR_CAPID0_A_0_0_0_PCI) — Offset E4h 157
2.32 Capability ID0 B (B_CR_CAPID0_B_0_0_0_PCI) — Offset E8h 159
2.33 Design and Engineering Backup Register 0 (DEBUP0_0_0_0_PCI) — Offset F4h 159
2.34 Design and Engineering Backup Register 1 (DEBUP1_0_0_0_PCI) — Offset FCh 160

3 Host Memory Mapped Configuration Space (MCHBAR) Registers **161**

3.1 DRAM Rank Population 0 (D_CR_DRP0) — Offset 1000h 161
3.2 DRAM Timing Register 9A (D_CR_DTR9A) — Offset 1004h 163
3.3 DRAM Timing Register 0A (D_CR_DTR0A) — Offset 1008h 163

3.4 DRAM Timing Register 1A (D_CR_DTR1A) — Offset 100Ch 164
3.5 DRAM Timing Register 2A (D_CR_DTR2A) — Offset 1010h..... 165
3.6 DRAM Timing Register 3A (D_CR_DTR3A) — Offset 1014h..... 165
3.7 DRAM Timing Register 4A (D_CR_DTR4A) — Offset 1018h..... 166
3.8 DRAM Timing Register 5A (D_CR_DTR5A) — Offset 101Ch 167
3.9 DRAM Timing Register 6A (D_CR_DTR6A) — Offset 1020h..... 168
3.10 DRAM Timing Register 7A (D_CR_DTR7A) — Offset 1024h..... 168
3.11 DRAM Timing Register 8A (D_CR_DTR8A) — Offset 1028h..... 169
3.12 D-Unit ODT Control Register A (D_CR_DOCRA) — Offset 102Ch 170
3.13 D-Unit Power Management Control 0 (D_CR_DPMC0) — Offset 1030h..... 171
3.14 D-Unit Power Management Control 1 (D_CR_DPMC1) — Offset 1034h..... 171
3.15 DRAM Refresh Control (D_CR_DRFC) — Offset 1038h 173
3.16 D-Unit Scheduler Control (D_CR_DSCH) — Offset 103Ch..... 174
3.17 DRAM Calibration Control (D_CR_DCAL) — Offset 1040h..... 176
3.18 DRAM Mode Registers Shadow Copy (D_CR_MR_SHADOW) — Offset 1048h 176
3.19 VNN Scaling Timer Control (D_CR_VNNTIMER) — Offset 104Ch..... 177
3.20 Periodic DRAM Temperature Polling Control (TQ) (D_CR_TQCTL) — Offset 1050h 177
3.21 Temperature Offset Control (D_CR_TQOFFSET) — Offset 1054h..... 178
3.22 D-Unit Control Operations (D_CR_DCO) — Offset 1058h 179
3.23 Data Scrambler (D_CR_SCRAMCTRL) — Offset 10A4h..... 180
3.24 Error Injection Address Register (D_CR_ERR_INJ) — Offset 10ACh..... 180
3.25 Error Injection Control Register (D_CR_ERR_INJ_CTL) — Offset 10B0h 181
3.26 Error Log Register (D_CR_ERR_ECC_LOG) — Offset 10B4h..... 181
3.27 D-Unit Fuse Status (D_CR_DFUSESTAT) — Offset 10BCh 182
3.28 Major Mode Control (D_CR_MMC) — Offset 1124h..... 183
3.29 Major Mode RD/WR Counter (Set A and B) (D_CR_MMRDWR_AB) — Offset 1128h..... 183
3.30 Major Mode RD/WR Counter (Set C and D) (D_CR_MMRDWR_CD) — Offset 112Ch 184
3.31 Access Class Initial Priority (D_CR_ACCIP) — Offset 1130h..... 184
3.32 Access Class 0 Priority Promotion Control (D_CR_RD_PROM0) — Offset 1134h..... 185
3.33 Access Class 1 Priority Promotion Control (D_CR_RD_PROM1) — Offset 1138h..... 186
3.34 Access Class 2 Priority Promotion Control (D_CR_RD_PROM2) — Offset 113Ch..... 186
3.35 Access Class 3 Priority Promotion Control (D_CR_RD_PROM3) — Offset 1140h..... 187
3.36 Access Class 4 Priority Promotion Control (D_CR_RD_PROM4) — Offset 1144h..... 187
3.37 Deadline Threshold (D_CR_DL_THRS) — Offset 1148h..... 188
3.38 Major Mode Blocking Rules Control (D_CR_MM_BLK) — Offset 114Ch 188
3.39 DRAM Self-Refresh Command (D_CR_DRAM_SR_CMD) — Offset 1154h 190
3.40 DQS Retraining Control (D_CR_DQS_RETRAINING_CTL) — Offset 1180h..... 190
3.41 MR4 De-Swizzle Control (D_CR_MR4_DESWIZZLE) — Offset 1184h..... 191
3.42 DRAM Rank Population 0 (D_CR_DRP0) — Offset 1200h..... 192
3.43 DRAM Timing Register 9A (D_CR_DTR9A) — Offset 1204h..... 194
3.44 DRAM Timing Register 0A (D_CR_DTR0A) — Offset 1208h..... 194
3.45 DRAM Timing Register 1A (D_CR_DTR1A) — Offset 120Ch 195
3.46 DRAM Timing Register 2A (D_CR_DTR2A) — Offset 1210h..... 196
3.47 DRAM Timing Register 3A (D_CR_DTR3A) — Offset 1214h..... 196
3.48 DRAM Timing Register 4A (D_CR_DTR4A) — Offset 1218h..... 197
3.49 DRAM Timing Register 5A (D_CR_DTR5A) — Offset 121Ch 198
3.50 DRAM Timing Register 6A (D_CR_DTR6A) — Offset 1220h..... 199
3.51 DRAM Timing Register 7A (D_CR_DTR7A) — Offset 1224h..... 199
3.52 DRAM Timing Register 8A (D_CR_DTR8A) — Offset 1228h..... 200
3.53 D-Unit ODT Control Register A (D_CR_DOCRA) — Offset 122Ch 201
3.54 D-Unit Power Management Control 0 (D_CR_DPMC0) — Offset 1230h..... 202
3.55 D-Unit Power Management Control 1 (D_CR_DPMC1) — Offset 1234h..... 202
3.56 DRAM Refresh Control (D_CR_DRFC) — Offset 1238h 204
3.57 D-Unit Scheduler Control (D_CR_DSCH) — Offset 123Ch..... 205
3.58 DRAM Calibration Control (D_CR_DCAL) — Offset 1240h..... 207

3.59 DRAM Mode Registers Shadow Copy (D_CR_MR_SHADOW) — Offset 1248h 207
3.60 VNN Scaling Timer Control (D_CR_VNNTIMER) — Offset 124Ch 208
3.61 Periodic DRAM Temperature Polling Control (TQ) (D_CR_TQCTL) — Offset 1250h 208
3.62 Temperature Offset Control (D_CR_TQOFFSET) — Offset 1254h 209
3.63 D-Unit Control Operations (D_CR_DCO) — Offset 1258h 210
3.64 Data Scrambler (D_CR_SCRAMCTRL) — Offset 12A4h 211
3.65 Error Injection Address Register (D_CR_ERR_INJ) — Offset 12ACh 211
3.66 Error Injection Control Register (D_CR_ERR_INJ_CTL) — Offset 12B0h 212
3.67 Error Log Register (D_CR_ERR_ECC_LOG) — Offset 12B4h 212
3.68 D-Unit Fuse Status (D_CR_DFUSESTAT) — Offset 12BCh 213
3.69 Major Mode Control (D_CR_MMC) — Offset 1324h 214
3.70 Major Mode RD/WR Counter (Set A and B) (D_CR_MMRDWR_AB) — Offset 1328h ... 214
3.71 Major Mode RD/WR Counter (Set C and D) (D_CR_MMRDWR_CD) — Offset 132Ch... 215
3.72 Access Class Initial Priority (D_CR_ACCIP) — Offset 1330h 215
3.73 Access Class 0 Priority Promotion Control (D_CR_RD_PROM0) — Offset 1334h 216
3.74 Access Class 1 Priority Promotion Control (D_CR_RD_PROM1) — Offset 1338h 217
3.75 Access Class 2 Priority Promotion Control (D_CR_RD_PROM2) — Offset 133Ch 217
3.76 Access Class 3 Priority Promotion Control (D_CR_RD_PROM3) — Offset 1340h 218
3.77 Access Class 4 Priority Promotion Control (D_CR_RD_PROM4) — Offset 1344h 218
3.78 Deadline Threshold (D_CR_DL_THRS) — Offset 1348h 219
3.79 Major Mode Blocking Rules Control (D_CR_MM_BLK) — Offset 134Ch 219
3.80 DRAM Self-Refresh Command (D_CR_DRAM_SR_CMD) — Offset 1354h 221
3.81 DQS Retraining Control (D_CR_DQS_RETRAINING_CTL) — Offset 1380h 221
3.82 MR4 De-Swizzle Control (D_CR_MR4_DESWIZZLE) — Offset 1384h 222
3.83 DRAM Rank Population 0 (D_CR_DRP0) — Offset 1400h 223
3.84 DRAM Timing Register 9A (D_CR_DTR9A) — Offset 1404h 225
3.85 DRAM Timing Register 0A (D_CR_DTR0A) — Offset 1408h 225
3.86 DRAM Timing Register 1A (D_CR_DTR1A) — Offset 140Ch 226
3.87 DRAM Timing Register 2A (D_CR_DTR2A) — Offset 1410h 227
3.88 DRAM Timing Register 3A (D_CR_DTR3A) — Offset 1414h 227
3.89 DRAM Timing Register 4A (D_CR_DTR4A) — Offset 1418h 228
3.90 DRAM Timing Register 5A (D_CR_DTR5A) — Offset 141Ch 229
3.91 DRAM Timing Register 6A (D_CR_DTR6A) — Offset 1420h 230
3.92 DRAM Timing Register 7A (D_CR_DTR7A) — Offset 1424h 230
3.93 DRAM Timing Register 8A (D_CR_DTR8A) — Offset 1428h 231
3.94 D-Unit ODT Control Register A (D_CR_DOCRA) — Offset 142Ch 232
3.95 D-Unit Power Management Control 0 (D_CR_DPMC0) — Offset 1430h 233
3.96 D-Unit Power Management Control 1 (D_CR_DPMC1) — Offset 1434h 233
3.97 DRAM Refresh Control (D_CR_DRFC) — Offset 1438h 235
3.98 D-Unit Scheduler Control (D_CR_DSCH) — Offset 143Ch 236
3.99 DRAM Calibration Control (D_CR_DCAL) — Offset 1440h 238
3.100 DRAM Mode Registers Shadow Copy (D_CR_MR_SHADOW) — Offset 1448h 238
3.101 VNN Scaling Timer Control (D_CR_VNNTIMER) — Offset 144Ch 239
3.102 Periodic DRAM Temperature Polling Control (TQ) (D_CR_TQCTL) — Offset 1450h 239
3.103 Temperature Offset Control (D_CR_TQOFFSET) — Offset 1454h 240
3.104 D-Unit Control Operations (D_CR_DCO) — Offset 1458h 241
3.105 Data Scrambler (D_CR_SCRAMCTRL) — Offset 14A4h 242
3.106 Error Injection Address Register (D_CR_ERR_INJ) — Offset 14ACh 242
3.107 Error Injection Control Register (D_CR_ERR_INJ_CTL) — Offset 14B0h 243
3.108 Error Log Register (D_CR_ERR_ECC_LOG) — Offset 14B4h 243
3.109 D-Unit Fuse Status (D_CR_DFUSESTAT) — Offset 14BCh 244
3.110 Major Mode Control (D_CR_MMC) — Offset 1524h 245
3.111 Major Mode RD/WR Counter (Set A and B) (D_CR_MMRDWR_AB) — Offset 1528h ... 245
3.112 Major Mode RD/WR Counter (Set C and D) (D_CR_MMRDWR_CD) — Offset 152Ch... 246
3.113 Access Class Initial Priority (D_CR_ACCIP) — Offset 1530h 246

3.114 Access Class 0 Priority Promotion Control (D_CR_RD_PROM0) — Offset 1534h 247
3.115 Access Class 1 Priority Promotion Control (D_CR_RD_PROM1) — Offset 1538h 248
3.116 Access Class 2 Priority Promotion Control (D_CR_RD_PROM2) — Offset 153Ch 248
3.117 Access Class 3 Priority Promotion Control (D_CR_RD_PROM3) — Offset 1540h 249
3.118 Access Class 4 Priority Promotion Control (D_CR_RD_PROM4) — Offset 1544h 249
3.119 Deadline Threshold (D_CR_DL_THRS) — Offset 1548h 250
3.120 Major Mode Blocking Rules Control (D_CR_MM_BLK) — Offset 154Ch 250
3.121 DRAM Self-Refresh Command (D_CR_DRAM_SR_CMD) — Offset 1554h 252
3.122 DQS Retraining Control (D_CR_DQS_RETRAINING_CTL) — Offset 1580h 252
3.123 MR4 De-Swizzle Control (D_CR_MR4_DESWIZZLE) — Offset 1584h 253
3.124 DRAM Rank Population 0 (D_CR_DRP0) — Offset 1600h 254
3.125 DRAM Timing Register 9A (D_CR_DTR9A) — Offset 1604h 256
3.126 DRAM Timing Register 0A (D_CR_DTR0A) — Offset 1608h 256
3.127 DRAM Timing Register 1A (D_CR_DTR1A) — Offset 160Ch 257
3.128 DRAM Timing Register 2A (D_CR_DTR2A) — Offset 1610h 258
3.129 DRAM Timing Register 3A (D_CR_DTR3A) — Offset 1614h 258
3.130 DRAM Timing Register 4A (D_CR_DTR4A) — Offset 1618h 259
3.131 DRAM Timing Register 5A (D_CR_DTR5A) — Offset 161Ch 260
3.132 DRAM Timing Register 6A (D_CR_DTR6A) — Offset 1620h 261
3.133 DRAM Timing Register 7A (D_CR_DTR7A) — Offset 1624h 261
3.134 DRAM Timing Register 8A (D_CR_DTR8A) — Offset 1628h 262
3.135 D-Unit ODT Control Register A (D_CR_DOCRA) — Offset 162Ch 263
3.136 D-Unit Power Management Control 0 (D_CR_DPMC0) — Offset 1630h 264
3.137 D-Unit Power Management Control 1 (D_CR_DPMC1) — Offset 1634h 264
3.138 DRAM Refresh Control (D_CR_DRFC) — Offset 1638h 266
3.139 D-Unit Scheduler Control (D_CR_DSCH) — Offset 163Ch 267
3.140 DRAM Calibration Control (D_CR_DCAL) — Offset 1640h 269
3.141 DRAM Mode Registers Shadow Copy (D_CR_MR_SHADOW) — Offset 1648h 269
3.142 VNN Scaling Timer Control (D_CR_VNNTIMER) — Offset 164Ch 270
3.143 Periodic DRAM Temperature Polling Control (TQ) (D_CR_TQCTL) — Offset 1650h 270
3.144 Temperature Offset Control (D_CR_TQOFFSET) — Offset 1654h 271
3.145 D-Unit Control Operations (D_CR_DCO) — Offset 1658h 272
3.146 Data Scrambler (D_CR_SCRAMCTRL) — Offset 16A4h 273
3.147 Error Injection Address Register (D_CR_ERR_INJ) — Offset 16ACh 273
3.148 Error Injection Control Register (D_CR_ERR_INJ_CTL) — Offset 16B0h 274
3.149 Error Log Register (D_CR_ERR_ECC_LOG) — Offset 16B4h 274
3.150 D-Unit Fuse Status (D_CR_DFUSESTAT) — Offset 16BCh 275
3.151 Major Mode Control (D_CR_MMC) — Offset 1724h 276
3.152 Major Mode RD/WR Counter (Set A and B) (D_CR_MMRDWR_AB) — Offset 1728h 276
3.153 Major Mode RD/WR Counter (Set C and D) (D_CR_MMRDWR_CD) — Offset 172Ch 277
3.154 Access Class Initial Priority (D_CR_ACCIP) — Offset 1730h 277
3.155 Access Class 0 Priority Promotion Control (D_CR_RD_PROM0) — Offset 1734h 278
3.156 Access Class 1 Priority Promotion Control (D_CR_RD_PROM1) — Offset 1738h 279
3.157 Access Class 2 Priority Promotion Control (D_CR_RD_PROM2) — Offset 173Ch 279
3.158 Access Class 3 Priority Promotion Control (D_CR_RD_PROM3) — Offset 1740h 280
3.159 Access Class 4 Priority Promotion Control (D_CR_RD_PROM4) — Offset 1744h 280
3.160 Deadline Threshold (D_CR_DL_THRS) — Offset 1748h 281
3.161 Major Mode Blocking Rules Control (D_CR_MM_BLK) — Offset 174Ch 281
3.162 DRAM Self-Refresh Command (D_CR_DRAM_SR_CMD) — Offset 1754h 283
3.163 DQS Retraining Control (D_CR_DQS_RETRAINING_CTL) — Offset 1780h 283
3.164 MR4 De-Swizzle Control (D_CR_MR4_DESWIZZLE) — Offset 1784h 284
3.165 DRAM Rank Population 0 (D_CR_DRP0) — Offset 1A00h 285
3.166 DRAM Timing Register 9A (D_CR_DTR9A) — Offset 1A04h 287
3.167 DRAM Timing Register 0A (D_CR_DTR0A) — Offset 1A08h 287
3.168 DRAM Timing Register 1A (D_CR_DTR1A) — Offset 1A0Ch 288

3.169 DRAM Timing Register 2A (D_CR_DTR2A) — Offset 1A10h 289
3.170 DRAM Timing Register 3A (D_CR_DTR3A) — Offset 1A14h 289
3.171 DRAM Timing Register 4A (D_CR_DTR4A) — Offset 1A18h 290
3.172 DRAM Timing Register 5A (D_CR_DTR5A) — Offset 1A1Ch 291
3.173 DRAM Timing Register 6A (D_CR_DTR6A) — Offset 1A20h 292
3.174 DRAM Timing Register 7A (D_CR_DTR7A) — Offset 1A24h 292
3.175 DRAM Timing Register 8A (D_CR_DTR8A) — Offset 1A28h 293
3.176 D-Unit ODT Control Register A (D_CR_DOCRA) — Offset 1A2Ch..... 294
3.177 D-Unit Power Management Control 0 (D_CR_DPMC0) — Offset 1A30h 295
3.178 D-Unit Power Management Control 1 (D_CR_DPMC1) — Offset 1A34h 295
3.179 DRAM Refresh Control (D_CR_DRFC) — Offset 1A38h 297
3.180 D-Unit Scheduler Control (D_CR_DSCH) — Offset 1A3Ch 298
3.181 DRAM Calibration Control (D_CR_DCAL) — Offset 1A40h 300
3.182 DRAM Mode Registers Shadow Copy (D_CR_MR_SHADOW) — Offset 1A48h 300
3.183 VNN Scaling Timer Control (D_CR_VNNTIMER) — Offset 1A4Ch 301
3.184 Periodic DRAM Temperature Polling Control (TQ) (D_CR_TQCTL) — Offset 1A50h..... 301
3.185 Temperature Offset Control (D_CR_TQOFFSET) — Offset 1A54h 302
3.186 D-Unit Control Operations (D_CR_DCO) — Offset 1A58h..... 303
3.187 Data Scrambler (D_CR_SCRAMCTRL) — Offset 1AA4h 304
3.188 Error Injection Address Register (D_CR_ERR_INJ) — Offset 1AACh 304
3.189 Error Injection Control Register (D_CR_ERR_INJ_CTL) — Offset 1AB0h..... 305
3.190 Error Log Register (D_CR_ERR_ECC_LOG) — Offset 1AB4h 305
3.191 D-Unit Fuse Status (D_CR_DFUSESTAT) — Offset 1ABCh..... 306
3.192 Major Mode Control (D_CR_MMC) — Offset 1B24h 307
3.193 Major Mode RD/WR Counter (Set A and B) (D_CR_MMRDWR_AB) — Offset 1B28h ... 307
3.194 Major Mode RD/WR Counter (Set C and D) (D_CR_MMRDWR_CD) — Offset 1B2Ch .. 308
3.195 Access Class Initial Priority (D_CR_ACCIP) — Offset 1B30h 308
3.196 Access Class 0 Priority Promotion Control (D_CR_RD_PROM0) — Offset 1B34h 309
3.197 Access Class 1 Priority Promotion Control (D_CR_RD_PROM1) — Offset 1B38h 310
3.198 Access Class 2 Priority Promotion Control (D_CR_RD_PROM2) — Offset 1B3Ch 310
3.199 Access Class 3 Priority Promotion Control (D_CR_RD_PROM3) — Offset 1B40h 311
3.200 Access Class 4 Priority Promotion Control (D_CR_RD_PROM4) — Offset 1B44h 311
3.201 Deadline Threshold (D_CR_DL_THRS) — Offset 1B48h 312
3.202 Major Mode Blocking Rules Control (D_CR_MM_BLK) — Offset 1B4Ch..... 312
3.203 DRAM Self-Refresh Command (D_CR_DRAM_SR_CMD) — Offset 1B54h..... 314
3.204 DQS Retraining Control (D_CR_DQS_RETRAINING_CTL) — Offset 1B80h 314
3.205 MR4 De-Swizzle Control (D_CR_MR4_DESWIZZLE) — Offset 1B84h 315
3.206 Security Control Policy (C_CR_SECURITY_CP_0_0_0_MCHBAR) — Offset 6000h 316
3.207 Security Read Access Control Policy (C_CR_SECURITY_RAC_0_0_0_MCHBAR) — Offset 6008h..... 316
3.208 Security Write Access Control Policy (C_CR_SECURITY_WAC_0_0_0_MCHBAR) — Offset 6010h..... 317
3.209 C_CR_BIOSWR_CP (C_CR_BIOSWR_CP_0_0_0_MCHBAR) — Offset 6018h 317
3.210 BIOSWR Read Access Control (C_CR_BIOSWR_RAC_0_0_0_MCHBAR) — Offset 6020h... 317
3.211 C_CR_BIOSWR_WAC (C_CR_BIOSWR_WAC_0_0_0_MCHBAR) — Offset 6028h..... 318
3.212 P_U_CODEWR_ALLRD Control Policy (C_CR_P_U_CODEWR_ALLRD_CP_0_0_0_MCHBAR) — Offset 6030h..... 318
3.213 P_U_CODEWR_ALLRD Read Access Control Policy (C_CR_P_U_CODEWR_ALL-RD_RAC_0_0_0_MCHBAR) — Offset 6038h..... 318
3.214 P_U_CODEWR_ALLRD Write Access Control Policy (C_CR_P_U_CODEWR_ALL-RD_WAC_0_0_0_MCHBAR) — Offset 6040h..... 319
3.215 BIOS/PMC WR Control Policy (C_CR_BIOS_PMC_WR_CP_0_0_0_MCHBAR) — Offset 6050h..... 319
3.216 BIOS/PMC WR Read Access Control (C_CR_BIOS_PMC_WR_RAC_0_0_0_MCHBAR) — Off-

set 6058h 320
3.217 BIOS/PMC WR Write Access Control (C_CR_BIOS_PMC_WR_WAC_0_0_0_MCHBAR) — Offset 6060h 320
3.218 P_U_PMC_CODEWR_ALLRD Control Policy (C_CR_P_U_PMC_CODEWR_ALL-RD_CP_0_0_0_MCHBAR) — Offset 6070h 320
3.219 P_U_PMC_CODEWR_ALLRD Read Access Control Policy (C_CR_P_U_PMC_CODEWR_ALL-RD_RAC_0_0_0_MCHBAR) — Offset 6078h 321
3.220 P_U_PMC_CODEWR_ALLRD Write Access Control Policy (C_CR_P_U_PMC_CODEWR_ALL-RD_WAC_0_0_0_MCHBAR) — Offset 6080h 321
3.221 Upstream Device Arbiter Grant Count A2T (A_CR_UPARB_GCNT_DEV_A2T_MCHBAR) — Offset 6400h 321
3.222 Upstream A2B Arbiter Channel 0 Grant Count (A_CR_UPARB_GCNT_A2B_0_MCHBAR) — Offset 6404h 322
3.223 Upstream A2B Arbiter Channel 1 Grant Count (A_CR_UPARB_GCNT_A2B_1_MCHBAR) — Offset 6408h 323
3.224 Upstream A2B Arbiter Channel 2 Grant Count (A_CR_UPARB_GCNT_A2B_2_MCHBAR) — Offset 640Ch 323
3.225 Upstream A2B Arbiter Channel 3 Grant Count (A_CR_UPARB_GCNT_A2B_3_MCHBAR) — Offset 6410h 324
3.226 Upstream A2B Arbiter Channel 4 Grant Count (A_CR_UPARB_GCNT_A2B_4_MCHBAR) — Offset 6414h 325
3.227 Upstream A2B Arbiter Channel 5 Grant Count (A_CR_UPARB_GCNT_A2B_5_MCHBAR) — Offset 6418h 325
3.228 Upstream A2B Arbiter Channel 6 Grant Count (A_CR_UPARB_GCNT_A2B_6_MCHBAR) — Offset 641Ch 326
3.229 Upstream A2B Arbiter Channel 7 Grant Count (A_CR_UPARB_GCNT_A2B_7_MCHBAR) — Offset 6420h 326
3.230 Upstream A2T Arbiter Channel 0 Grant Count (A_CR_UPARB_GCNT_A2T_0_MCHBAR) — Offset 6424h 327
3.231 Upstream P2P Arbiter Channel 0 Grant Count (A_CR_UPARB_GCNT_P2P_0_MCHBAR) — Offset 6428h 328
3.232 Upstream P2P Arbiter Channel 1 Grant Count (A_CR_UPARB_GCNT_P2P_1_MCHBAR) — Offset 642Ch 328
3.233 Upstream Private Credit Return Grant Count Posted 0 (A_CR_CRDAR-B_PRIV_GCNT_DEV_P_0_MCHBAR) — Offset 6430h 329
3.234 Upstream Private Credit Return Grant Count Posted 1 (A_CR_CRDAR-B_PRIV_GCNT_DEV_P_1_MCHBAR) — Offset 6434h 329
3.235 Upstream Private Credit Return Grant Count Non-posted 0 (A_CR_CRDAR-B_PRIV_GCNT_DEV_N_0_MCHBAR) — Offset 6438h 330
3.236 Upstream Private Credit Return Grant Count Posted 1 (A_CR_CRDAR-B_PRIV_GCNT_DEV_N_1_MCHBAR) — Offset 643Ch 331
3.237 Upstream Private Credit Return Grant Count Completion (A_CR_CRDAR-B_PRIV_GCNT_DEV_C_0_MCHBAR) — Offset 6440h 331
3.238 Upstream Shared Credit Return Grant Count Posted 0 (A_CR_CRDAR-B_SHRD_GCNT_DEV_P_0_MCHBAR) — Offset 6444h 332
3.239 Upstream Shared Credit Return Grant Count Posted 1 (A_CR_CRDAR-B_SHRD_GCNT_DEV_P_1_MCHBAR) — Offset 6448h 333
3.240 Upstream Shared Credit Return Grant Count Non-posted 0 (A_CR_CRDAR-B_SHRD_GCNT_DEV_N_0_MCHBAR) — Offset 644Ch 333
3.241 Upstream Shared Credit Return Grant Count Posted 1 (A_CR_CRDAR-B_SHRD_GCNT_DEV_N_1_MCHBAR) — Offset 6450h 334
3.242 Upstream Shared Credit Return Grant Count Completion (A_CR_CRDAR-B_SHRD_GCNT_DEV_C_0_MCHBAR) — Offset 6454h 334
3.243 Upstream Credit Arbiter Private Credit Return Class Arbiter Grant Count (A_CR_CRDAR-B_PRIV_GCNT_CLS_MCHBAR) — Offset 6458h 335
3.244 Upstream Credit Arbiter Shared Cedit Return Class Arbiter Grant Count (A_CR_CRDAR-B_SHRD_GCNT_CLS_MCHBAR) — Offset 645Ch 336

3.245 Gazelle Queue Limit Channel 0-3 (A_CR_GZLQ_LIMIT_CH0_3_MCHBAR) — Offset 6460h 336
3.246 Gazelle Queue Limit Channels 4-7 (A_CR_GZLQ_LIMIT_CH4_7_MCHBAR) — Offset 6464h 337
3.247 IOMMU Arbiter Grant Count VC0a Register (A_CR_IOMMUARB_GCNT_VC0A_0_0_0_MCHBAR) — Offset 6468h 337
3.248 IOMMU Arbiter Grant Count VC0b Register (A_CR_IOMMUARB_GCNT_VC0B_0_0_0_MCHBAR) — Offset 646Ch 338
3.249 IOMMU Arbiter Grant Count VC1b Register (A_CR_IOMMUARB_GCNT_VC1B_0_0_0_MCHBAR) — Offset 6470h 338
3.250 Gazelle Queue Reserved Entries Channels 0-3 (A_CR_GZLQ_RSVD_CH0_3_MCHBAR) — Offset 6474h 339
3.251 Gazelle Queue Reserved Entries Channels 4-7 (A_CR_GZLQ_RSVD_CH4_7_MCHBAR) — Offset 6478h 339
3.252 Spare BIOS (A_CR_SPARE_BIOS_MCHBAR) — Offset 647Ch 340
3.253 Upcmd Credit Maximum Channel 0 (A_CR_UPCMD_CRDTMAX_CH0_0_0_0_MCHBAR) — Offset 6490h 340
3.254 Upcmd Credit Maximum Channel 1 (A_CR_UPCMD_CRDTMAX_CH1_0_0_0_MCHBAR) — Offset 6494h 341
3.255 Upcmd Credit Maximum Channel 2 (A_CR_UPCMD_CRDTMAX_CH2_0_0_0_MCHBAR) — Offset 6498h 341
3.256 Upcmd Credit Maximum Channel 3 (A_CR_UPCMD_CRDTMAX_CH3_0_0_0_MCHBAR) — Offset 649Ch 342
3.257 Upcmd Credit Maximum Channel 4 (A_CR_UPCMD_CRDTMAX_CH4_0_0_0_MCHBAR) — Offset 64A0h 342
3.258 Upcmd Credit Maximum Channel 5 (A_CR_UPCMD_CRDTMAX_CH5_0_0_0_MCHBAR) — Offset 64A4h 343
3.259 Upcmd Credit Maximum Channel 6 (A_CR_UPCMD_CRDTMAX_CH6_0_0_0_MCHBAR) — Offset 64A8h 343
3.260 Upcmd Credit Maximum Channel 7 (A_CR_UPCMD_CRDTMAX_CH7_0_0_0_MCHBAR) — Offset 64ACh 344
3.261 MOT OUT Base Register (A_CR_MOT_OUT_BASE_0_0_0_MCHBAR) — Offset 64C0h . 344
3.262 MOT OUT Mask Register (A_CR_MOT_OUT_MASK_0_0_0_MCHBAR) — Offset 64C4h 345
3.263 A-Unit BIOSWR Control Policy (A_CR_BIOSWR_CP_0_0_0_MCHBAR) — Offset 64C8h 346
3.264 A-Unit BIOSWR Ready Access Control (A_CR_BIOSWR_RAC_0_0_0_MCHBAR) — Offset 64D0h 346
3.265 BIOSWR Write Access Control (A_CR_BIOSWR_WAC_0_0_0_MCHBAR) — Offset 64D8h . 346
3.266 AUnit Pcode/Ucode Write, All Read Control Policy Register (A_CR_P_U_CODEWR_ALLRD_CP_0_0_0_MCHBAR) — Offset 64E0h 347
3.267 AUnit Pcode/Ucode Write, All Read Read Access Control Policy Register (A_CR_P_U_CODEWR_ALLRD_RAC_0_0_0_MCHBAR) — Offset 64E8h 347
3.268 AUnit Pcode/Ucode Write, All Read Write Access Control Policy Register (A_CR_P_U_CODEWR_ALLRD_WAC_0_0_0_MCHBAR) — Offset 64F0h 348
3.269 CHAP Select 1 (A_CR_CHAP_SLCT1_MCHBAR) — Offset 6500h..... 348
3.270 CHAP Select 2 (A_CR_CHAP_SLCT2_MCHBAR) — Offset 6504h..... 349
3.271 CHAP Select 3 (A_CR_CHAP_SLCT3_MCHBAR) — Offset 6508h..... 349
3.272 A_IMRGLOBAL_BM Control Policy (A_CR_IMRGLOBAL_BM_CP_0_0_0_MCHBAR) — Offset 6510h..... 350
3.273 A_IMRGLOBAL_BM Read Access Control (A_CR_IMRGLOBAL_BM_RAC_0_0_0_MCHBAR) — Offset 6518h..... 350
3.274 A_IMRGLOBAL_BM Write Access Control (A_CR_IMRGLOBAL_BM_WAC_0_0_0_MCHBAR) — Offset 6520h..... 350
3.275 Uncorrectable Error Status Register (A_CR_UNCERRSTS_0_0_0_MCHBAR) — Offset 6588h..... 351
3.276 Uncorrectable Error Mask Register (A_CR_UNCERRMSK_0_0_0_MCHBAR) — Offset 658Ch

353
3.277 Slice and Channel Hash (A_CR_SLICE_CHANNEL_HASH_0_0_0_MCHBAR) — Offset 65C0h 354
3.278 Mirror Range Register (A_CR_MIRROR_RANGE_0_0_0_MCHBAR) — Offset 65C8h 356
3.279 ASYM MEM REGION 0 CONFIGURATION WITH NO INTERLEAVING (A_CR_ASYM_MEM_REGION0_0_0_0_MCHBAR) — Offset 65D0h 357
3.280 ASYM MEM REGION 1 CONFIGURATION WITH NO INTERLEAVING (A_CR_ASYM_MEM_REGION1_0_0_0_MCHBAR) — Offset 65D4h 357
3.281 Two-Way Asymmetric Memory Region Configuration (A_CR_ASYM_2WAY_MEM_REGION_0_0_0_MCHBAR) — Offset 65D8h 358
3.282 B-Unit Miscellaneous Configuration (B_CR_BMISC_0_0_0_MCHBAR) — Offset 6800h 359
3.283 Security Control Policy (B_CR_SECURITY_CP_0_0_0_MCHBAR) — Offset 6808h 359
3.284 Security Group Read Access Policy (B_CR_SECURITY_RAC_0_0_0_MCHBAR) — Offset 6810h 363
3.285 Security Group Write Access Policy (B_CR_SECURITY_WAC_0_0_0_MCHBAR) — Offset 6818h 367
3.286 Slice 0 Memory Access Count (B_CR_MEM_ACCESS_COUNT_SLICE0) — Offset 6868h ... 371
3.287 Slice 1 Memory Access Count (B_CR_MEM_ACCESS_COUNT_SLICE1) — Offset 686Ch ... 371
3.288 IMR0 Base (B_CR_BIMR0BASE_0_0_0_MCHBAR) — Offset 6870h 371
3.289 IMR0 Mask (B_CR_BIMR0MASK_0_0_0_MCHBAR) — Offset 6874h 372
3.290 IMR0 Control Policy (B_CR_BIMR0CP_0_0_0_MCHBAR) — Offset 6878h 372
3.291 IMR0 Read Access Policy (B_CR_BIMR0RAC_0_0_0_MCHBAR) — Offset 6880h 376
3.292 IMR0 Write Access Policy (B_CR_BIMR0WAC_0_0_0_MCHBAR) — Offset 6888h 380
3.293 IMR1 Base (B_CR_BIMR1BASE_0_0_0_MCHBAR) — Offset 6890h 384
3.294 IMR1 Mask (B_CR_BIMR1MASK_0_0_0_MCHBAR) — Offset 6894h 384
3.295 IMR1 Control Policy (B_CR_BIMR1CP_0_0_0_MCHBAR) — Offset 6898h 385
3.296 IMR1 Read Access Policy (B_CR_BIMR1RAC_0_0_0_MCHBAR) — Offset 68A0h 389
3.297 IMR1 Write Access Policy (B_CR_BIMR1WAC_0_0_0_MCHBAR) — Offset 68A8h 392
3.298 IMR2 Base (B_CR_BIMR2BASE_0_0_0_MCHBAR) — Offset 68B0h 396
3.299 IMR2 Mask (B_CR_BIMR2MASK_0_0_0_MCHBAR) — Offset 68B4h 397
3.300 IMR2 Control Policy (B_CR_BIMR2CP_0_0_0_MCHBAR) — Offset 68B8h 397
3.301 IMR2 Read Access Policy (B_CR_BIMR2RAC_0_0_0_MCHBAR) — Offset 68C0h 401
3.302 IMR2 Write Access Policy (B_CR_BIMR2WAC_0_0_0_MCHBAR) — Offset 68C8h 405
3.303 IMR3 Base (B_CR_BIMR3BASE_0_0_0_MCHBAR) — Offset 68D0h 408
3.304 IMR3 Mask (B_CR_BIMR3MASK_0_0_0_MCHBAR) — Offset 68D4h 409
3.305 IMR3 Control Policy (B_CR_BIMR3CP_0_0_0_MCHBAR) — Offset 68D8h 410
3.306 IMR3 Read Access Policy (B_CR_BIMR3RAC_0_0_0_MCHBAR) — Offset 68E0h..... 413
3.307 IMR3 Write Access Policy (B_CR_BIMR3WAC_0_0_0_MCHBAR) — Offset 68E8h 417
3.308 IMR4 Base (B_CR_BIMR4BASE_0_0_0_MCHBAR) — Offset 68F0h 421
3.309 IMR4 Mask (B_CR_BIMR4MASK_0_0_0_MCHBAR) — Offset 68F4h 421
3.310 IMR4 Control Policy (B_CR_BIMR4CP_0_0_0_MCHBAR) — Offset 68F8h 422
3.311 IMR4 Read Access Policy (B_CR_BIMR4RAC_0_0_0_MCHBAR) — Offset 6900h 426
3.312 IMR4 Write Access Policy (B_CR_BIMR4WAC_0_0_0_MCHBAR) — Offset 6908h 430
3.313 IMR5 Base (B_CR_BIMR5BASE_0_0_0_MCHBAR) — Offset 6910h 433
3.314 IMR5 Mask (B_CR_BIMR5MASK_0_0_0_MCHBAR) — Offset 6914h 434
3.315 IMR5 Control Policy (B_CR_BIMR5CP_0_0_0_MCHBAR) — Offset 6918h 435
3.316 IMR5 Read Access Policy (B_CR_BIMR5RAC_0_0_0_MCHBAR) — Offset 6920h 438
3.317 IMR5 Write Access Policy (B_CR_BIMR5WAC_0_0_0_MCHBAR) — Offset 6928h 442
3.318 IMR6 Base (B_CR_BIMR6BASE_0_0_0_MCHBAR) — Offset 6930h 446
3.319 IMR6 Mask (B_CR_BIMR6MASK_0_0_0_MCHBAR) — Offset 6934h 446
3.320 IMR6 Control Policy (B_CR_BIMR6CP_0_0_0_MCHBAR) — Offset 6938h 447
3.321 IMR6 Read Access Policy (B_CR_BIMR6RAC_0_0_0_MCHBAR) — Offset 6940h 451
3.322 IMR6 Write Access Policy (B_CR_BIMR6WAC_0_0_0_MCHBAR) — Offset 6948h 455

3.323 IMR7 Base (B_CR_BIMR7BASE_0_0_0_MCHBAR) — Offset 6950h 458
3.324 IMR7 Mask (B_CR_BIMR7MASK_0_0_0_MCHBAR) — Offset 6954h 459
3.325 IMR7 Control Policy (B_CR_BIMR7CP_0_0_0_MCHBAR) — Offset 6958h 460
3.326 IMR7 Read Access Policy (B_CR_BIMR7RAC_0_0_0_MCHBAR) — Offset 6960h 463
3.327 IMR7 Write Access Policy (B_CR_BIMR7WAC_0_0_0_MCHBAR) — Offset 6968h 467
3.328 IMR8 Base (B_CR_BIMR8BASE_0_0_0_MCHBAR) — Offset 6970h 471
3.329 IMR8 Mask (B_CR_BIMR8MASK_0_0_0_MCHBAR) — Offset 6974h 471
3.330 IMR8 Control Policy (B_CR_BIMR8CP_0_0_0_MCHBAR) — Offset 6978h 472
3.331 IMR8 Read Access Policy (B_CR_BIMR8RAC_0_0_0_MCHBAR) — Offset 6980h 476
3.332 IMR8 Write Access Policy (B_CR_BIMR8WAC_0_0_0_MCHBAR) — Offset 6988h 480
3.333 IMR9 Base (B_CR_BIMR9BASE_0_0_0_MCHBAR) — Offset 6990h 483
3.334 IMR9 Mask (B_CR_BIMR9MASK_0_0_0_MCHBAR) — Offset 6994h 484
3.335 IMR9 Control Policy (B_CR_BIMR9CP_0_0_0_MCHBAR) — Offset 6998h 485
3.336 IMR9 Read Access Policy (B_CR_BIMR9RAC_0_0_0_MCHBAR) — Offset 69A0h 488
3.337 IMR9 Write Access Policy (B_CR_BIMR9WAC_0_0_0_MCHBAR) — Offset 69A8h 492
3.338 IMR10 Base (B_CR_BIMR10BASE_0_0_0_MCHBAR) — Offset 69B0h 496
3.339 IMR10 Mask (B_CR_BIMR10MASK_0_0_0_MCHBAR) — Offset 69B4h 496
3.340 IMR10 Control Policy (B_CR_BIMR10CP_0_0_0_MCHBAR) — Offset 69B8h 497
3.341 IMR10 Read Access Policy (B_CR_BIMR10RAC_0_0_0_MCHBAR) — Offset 69C0h 501
3.342 IMR10 Write Access Policy (B_CR_BIMR10WAC_0_0_0_MCHBAR) — Offset 69C8h ... 505
3.343 IMR11 Base (B_CR_BIMR11BASE_0_0_0_MCHBAR) — Offset 69D0h 508
3.344 IMR11 Mask (B_CR_BIMR11MASK_0_0_0_MCHBAR) — Offset 69D4h 509
3.345 IMR11 Control Policy (B_CR_BIMR11CP_0_0_0_MCHBAR) — Offset 69D8h 510
3.346 IMR11 Read Access Policy (B_CR_BIMR11RAC_0_0_0_MCHBAR) — Offset 69E0h..... 513
3.347 IMR11 Write Access Policy (B_CR_BIMR11WAC_0_0_0_MCHBAR) — Offset 69E8h ... 517
3.348 IMR12 Base (B_CR_BIMR12BASE_0_0_0_MCHBAR) — Offset 69F0h 521
3.349 IMR12 Mask (B_CR_BIMR12MASK_0_0_0_MCHBAR) — Offset 69F4h 521
3.350 IMR12 Control Policy (B_CR_BIMR12CP_0_0_0_MCHBAR) — Offset 69F8h 522
3.351 IMR12 Read Access Policy (B_CR_BIMR12RAC_0_0_0_MCHBAR) — Offset 6A00h 526
3.352 IMR12 Write Access Policy (B_CR_BIMR12WAC_0_0_0_MCHBAR) — Offset 6A08h ... 530
3.353 IMR13 Base (B_CR_BIMR13BASE_0_0_0_MCHBAR) — Offset 6A10h 533
3.354 IMR13 Mask (B_CR_BIMR13MASK_0_0_0_MCHBAR) — Offset 6A14h 534
3.355 IMR13 Control Policy (B_CR_BIMR13CP_0_0_0_MCHBAR) — Offset 6A18h 535
3.356 IMR13 Read Access Policy (B_CR_BIMR13RAC_0_0_0_MCHBAR) — Offset 6A20h 538
3.357 IMR13 Write Access Policy (B_CR_BIMR13WAC_0_0_0_MCHBAR) — Offset 6A28h ... 542
3.358 IMR14 Base (B_CR_BIMR14BASE_0_0_0_MCHBAR) — Offset 6A30h 546
3.359 IMR14 Mask (B_CR_BIMR14MASK_0_0_0_MCHBAR) — Offset 6A34h 546
3.360 IMR14 Control Policy (B_CR_BIMR14CP_0_0_0_MCHBAR) — Offset 6A38h 547
3.361 IMR14 Read Access Policy (B_CR_BIMR14RAC_0_0_0_MCHBAR) — Offset 6A40h 551
3.362 IMR14 Write Access Policy (B_CR_BIMR14WAC_0_0_0_MCHBAR) — Offset 6A48h ... 555
3.363 IMR15 Base (B_CR_BIMR15BASE_0_0_0_MCHBAR) — Offset 6A50h 558
3.364 IMR15 Mask (B_CR_BIMR15MASK_0_0_0_MCHBAR) — Offset 6A54h 559
3.365 IMR15 Control Policy (B_CR_BIMR15CP_0_0_0_MCHBAR) — Offset 6A58h 560
3.366 IMR15 Read Access Policy (B_CR_BIMR15RAC_0_0_0_MCHBAR) — Offset 6A60h 563
3.367 IMR15 Write Access Policy (B_CR_BIMR15WAC_0_0_0_MCHBAR) — Offset 6A68h ... 567
3.368 IMR16 Base (B_CR_BIMR16BASE_0_0_0_MCHBAR) — Offset 6A70h 571
3.369 IMR16 Mask (B_CR_BIMR16MASK_0_0_0_MCHBAR) — Offset 6A74h 571
3.370 IMR16 Control Policy (B_CR_BIMR16CP_0_0_0_MCHBAR) — Offset 6A78h 572
3.371 IMR16 Read Access Policy (B_CR_BIMR16RAC_0_0_0_MCHBAR) — Offset 6A80h 576
3.372 IMR16 Write Access Policy (B_CR_BIMR16WAC_0_0_0_MCHBAR) — Offset 6A88h ... 580
3.373 IMR17 Base (B_CR_BIMR17BASE_0_0_0_MCHBAR) — Offset 6A90h 583
3.374 IMR17 Mask (B_CR_BIMR17MASK_0_0_0_MCHBAR) — Offset 6A94h 584
3.375 IMR17 Control Policy (B_CR_BIMR17CP_0_0_0_MCHBAR) — Offset 6A98h 585
3.376 IMR17 Read Access Policy (B_CR_BIMR17RAC_0_0_0_MCHBAR) — Offset 6AA0h 588
3.377 IMR17 Write Access Policy (B_CR_BIMR17WAC_0_0_0_MCHBAR) — Offset 6AA8h ... 592

3.378 IMR18 Base (B_CR_BIMR18BASE_0_0_0_MCHBAR) — Offset 6AB0h 596
3.379 IMR18 Mask (B_CR_BIMR18MASK_0_0_0_MCHBAR) — Offset 6AB4h 596
3.380 IMR18 Control Policy (B_CR_BIMR18CP_0_0_0_MCHBAR) — Offset 6AB8h 597
3.381 IMR18 Read Access Policy (B_CR_BIMR18RAC_0_0_0_MCHBAR) — Offset 6AC0h 601
3.382 IMR18 Write Access Policy (B_CR_BIMR18WAC_0_0_0_MCHBAR) — Offset 6AC8h ... 605
3.383 IMR19 Base (B_CR_BIMR19BASE_0_0_0_MCHBAR) — Offset 6AD0h 608
3.384 IMR19 Mask (B_CR_BIMR19MASK_0_0_0_MCHBAR) — Offset 6AD4h 609
3.385 IMR19 Control Policy (B_CR_BIMR19CP_0_0_0_MCHBAR) — Offset 6AD8h 610
3.386 IMR19 Read Access Policy (B_CR_BIMR19RAC_0_0_0_MCHBAR) — Offset 6AE0h 613
3.387 IMR19 Write Access Policy (B_CR_BIMR19WAC_0_0_0_MCHBAR) — Offset 6AE8h 617
3.388 MOT Out Base (B_CR_MOT_OUT_BASE_0_0_0_MCHBAR) — Offset 6AF0h 621
3.389 MOT Out Mask (B_CR_MOT_OUT_MASK_0_0_0_MCHBAR) — Offset 6AF4h 621
3.390 MOT Buffer Control Policy (B_CR_BMOT_BUF_CP_0_0_0_MCHBAR) — Offset 6AF8h . 622
3.391 MOT Buffer Read Access Policy (B_CR_BMOT_BUF_RAC_0_0_0_MCHBAR) — Offset 6B00h 626
3.392 MOT Buffer Write Access Policy (B_CR_BMOT_BUF_WAC_0_0_0_MCHBAR) — Offset 6B08h 630
3.393 IMR Global BM Control Policy (B_CR_BIMRGLOBAL_BM_CP_0_0_0_MCHBAR) — Offset 6B10h 633
3.394 IMR Global BM Read Access Control (B_CR_BIMRGLOBAL_BM_RAC_0_0_0_MCHBAR) — Offset 6B18h 637
3.395 IMR Global BM Write Access Policy (B_CR_BIMRGLOBAL_BM_WAC_0_0_0_MCHBAR) — Offset 6B20h 637
3.396 Graphics Stolen Memory Control Policy (B_CR_BGSMCP_0_0_0_MCHBAR) — Offset 6B28h 641
3.397 GSM Read Access Policy (B_CR_BGSMRAC_0_0_0_MCHBAR) — Offset 6B30h 645
3.398 GSM Write Access Policy (B_CR_BGSMWAC_0_0_0_MCHBAR) — Offset 6B38h 649
3.399 TPM Control Policy (B_CR_TPM_CP_0_0_0_MCHBAR) — Offset 6B40h 652
3.400 TPM Access Control (B_CR_TPM_AC_0_0_0_MCHBAR) — Offset 6B48h 656
3.401 BGSM Control Register (B_CR_BGSM_CTRL_0_0_0_MCHBAR) — Offset 6B50h 656
3.402 SMM Control Register (B_CR_BSMR_CTRL_0_0_0_MCHBAR) — Offset 6B54h 657
3.403 Default VTd Control Register (B_CR_BDEFVTDPMR_CTRL_0_0_0_MCHBAR) — Offset 6B58h 658
3.404 MOT Trigger Trace Control (B_CR_MOT_TRIG_TRACE_CTRL_0_0_0_MCHBAR) — Offset 6B7Ch 658
3.405 MOT Slice 0 Memory Pointer (B_CR_MOT_SLICE_0_MEM_PTR_0_0_0_MCHBAR) — Offset 6B80h 660
3.406 MOT Slice 1 Memory Pointer (B_CR_MOT_SLICE_1_MEM_PTR_0_0_0_MCHBAR) — Offset 6B88h 660
3.407 MOT Slice 0 Record ID (B_CR_MOT_SLICE_0_RECORD_ID_0_0_0_MCHBAR) — Offset 6B90h 661
3.408 MOT Slice 1 Record ID (B_CR_MOT_SLICE_1_RECORD_ID_0_0_0_MCHBAR) — Offset 6B94h 661
3.409 MOT Filter Match 0 (B_CR_MOT_FILTER_MATCH0_0_0_0_MCHBAR) — Offset 6BA0h 662
3.410 MOT Filter Mask (B_CR_MOT_FILTER_MASK0_0_0_0_MCHBAR) — Offset 6BA8h 662
3.411 MOT Filter Match 1 (B_CR_MOT_FILTER_MATCH1_0_0_0_MCHBAR) — Offset 6BB0h 663
3.412 MOT Filter Mask 1 (B_CR_MOT_FILTER_MASK1_0_0_0_MCHBAR) — Offset 6BB8h ... 663
3.413 MOT Filter Misc 0 (B_CR_MOT_FILTER_MISC0_0_0_0_MCHBAR) — Offset 6BC0h 664
3.414 MOT Filter Misc 1 (B_CR_MOT_FILTER_MISC1_0_0_0_MCHBAR) — Offset 6BC8h 664
3.415 MOT Trigger Match 0 (B_CR_MOT_TRIGGER_MATCH0_0_0_0_MCHBAR) — Offset 6BD0h 665
3.416 MOT Trigger Mask 0 (B_CR_MOT_TRIGGER_MASK0_0_0_0_MCHBAR) — Offset 6BD8h .. 665
3.417 MOT Trigger Match 1 (B_CR_MOT_TRIGGER_MATCH1_0_0_0_MCHBAR) — Offset 6BE0h 666
3.418 MOT Trigger Mask 1 (B_CR_MOT_TRIGGER_MASK1_0_0_0_MCHBAR) — Offset 6BE8h ..

666
3.419 MOT Trigger Misc 0 (B_CR_MOT_TRIGGER_MISC0_0_0_0_MCHBAR) — Offset 6BF0h 667
3.420 MOT Trigger Misc 1 (B_CR_MOT_TRIGGER_MISC1_0_0_0_MCHBAR) — Offset 6BF8h 667
3.421 MOT PSMI Sync (B_CR_MOT_PSMI_SYNC_0_0_0_MCHBAR) — Offset 6C00h 668
3.422 B_CR_BIOSWR_CP (B_CR_BIOSWR_CP_0_0_0_MCHBAR) — Offset 6C08h 668
3.423 BIOSWR Read Access Policy (B_CR_BIOSWR_RAC_0_0_0_MCHBAR) — Offset 6C10h 672
3.424 B_CR_BIOSWR_WAC (B_CR_BIOSWR_WAC_0_0_0_MCHBAR) — Offset 6C18h..... 675
3.425 Destination Override for BIOS1 region (B_CR_BIOS1_DECODE_OVERRIDE_0_0_0_MCHBAR) — Offset 6C20h..... 678
3.426 TPM Selector (B_CR_TPM_SELECTOR_0_0_0_MCHBAR) — Offset 6C24h..... 679
3.427 B-Unit Pcode/Ucode Write, All Read Control Policy Register (B_CR_P_U_CODEWR_ALLRD_CP_0_0_0_MCHBAR) — Offset 6C28h..... 679
3.428 B-Unit Pcode/Ucode Read Access Policy (B_CR_P_U_CODEWR_ALLRD_RAC_0_0_0_MCHBAR) — Offset 6C30h..... 684
3.429 B-Unit Pcode/Ucode Write Access Policy (B_CR_P_U_CODEWR_ALLRD_WAC_0_0_0_MCHBAR) — Offset 6C38h..... 688
3.430 Default VTd BAR (B_CR_DEFVTDBAR_0_0_0_MCHBAR) — Offset 6C80h 693
3.431 Gfx VTd BAR (B_CR_GFXVTDBAR_0_0_0_MCHBAR) — Offset 6C88h 694
3.432 B-Unit Lites Group 0 Control (B_CR_LITES0_CTL_0_0_0_MCHBAR) — Offset 6C90h . 694
3.433 B-Unit Lites Group 0 Opcode Match Filter (B_CR_LITES0_OPCODE_MATCH_0_0_0_MCHBAR) — Offset 6C94h..... 696
3.434 B-Unit Lites Group 0 Agent Match Filter (B_CR_LITES0_AGENT_MATCH_0_0_0_MCHBAR) — Offset 6C98h 697
3.435 B-Unit Lites Group 0 U2C IntData Match Filter (B_CR_LITES0_U2CINTDATA_MATCH_0_0_0_MCHBAR) — Offset 6C9Ch 698
3.436 B-Unit Lites Group 0 Address Match Filter LITES0_ADDR_MATCH (B_CR_LITES0_ADDR_MATCH_0_0_0_MCHBAR) — Offset 6CA0h 698
3.437 B-Unit Lites Group 0 Address Mask Filter LITES0_ADDR_MASK (B_CR_LITES0_ADDR_MASK_0_0_0_MCHBAR) — Offset 6CA8h 698
3.438 B-Unit Lites Group 0 Data Match Filter LITES0_DATA_MATCH (B_CR_LITES0_DATA_MATCH_0_0_0_MCHBAR) — Offset 6CB0h 699
3.439 B-Unit Lites Group 0 Data Mask Filter LITES0_DATA_MASK (B_CR_LITES0_DATA_MASK_0_0_0_MCHBAR) — Offset 6CB4h..... 700
3.440 B-Unit Lites Group 1 Control (B_CR_LITES1_CTL_0_0_0_MCHBAR) — Offset 6CC0h. 700
3.441 B-Unit Lites Group 1 Opcode Match Filter (B_CR_LITES1_OPCODE_MATCH_0_0_0_MCHBAR) — Offset 6CC4h 702
3.442 B-Unit Lites Group 1 Agent Match Filter (B_CR_LITES1_AGENT_MATCH_0_0_0_MCHBAR) — Offset 6CC8h 703
3.443 B-Unit Lites Group 1 U2C IntData Match Filter (B_CR_LITES1_U2CINTDATA_MATCH_0_0_0_MCHBAR) — Offset 6CCCh 704
3.444 B-Unit Lites Group 1 Address Match Filter LITES1_ADDR_MATCH (B_CR_LITES1_ADDR_MATCH_0_0_0_MCHBAR) — Offset 6CD0h 704
3.445 B-Unit Lites Group 1 Address Mask Filter LITES1_ADDR_MASK (B_CR_LITES1_ADDR_MASK_0_0_0_MCHBAR) — Offset 6CD8h 704
3.446 B-Unit Lites Group 1 Data Match Filter LITES1_DATA_MATCH (B_CR_LITES1_DATA_MATCH_0_0_0_MCHBAR) — Offset 6CE0h 705
3.447 B-Unit Lites Group 1 Data Mask Filter LITES1_DATA_MASK (B_CR_LITES1_DATA_MASK_0_0_0_MCHBAR) — Offset 6CE4h..... 706
3.448 B-Unit Lites Group 2 Control (B_CR_LITES2_CTL_0_0_0_MCHBAR) — Offset 6CF0h . 706
3.449 B-Unit Lites Group 2 Opcode Match Filter (B_CR_LITES2_OPCODE_MATCH_0_0_0_MCHBAR) — Offset 6CF4h..... 708
3.450 B-Unit Lites Group 2 Agent Match Filter (B_CR_LITES2_AGENT_MATCH_0_0_0_MCHBAR) — Offset 6CF8h..... 709
3.451 B-Unit Lites Group 2 U2C IntData Match Filter (B_CR_LITES2_U2CINTDATA_MATCH_0_0_0_MCHBAR) — Offset 6CFCh 710
3.452 B-Unit Lites Group 2 Address Match Filter LITES2_ADDR_MATCH (B_CR_LITES2_ADDR_MATCH_0_0_0_MCHBAR) — Offset 6D00h 710

3.453 B-Unit Lites Group 2 Address Mask Filter LITES2_ADDR_MASK (B_CR_LITES2_ADDR_MASK_0_0_0_MCHBAR) — Offset 6D08h 710
3.454 B-Unit Lites Group 2 Data Match Filter LITES2_DATA_MATCH (B_CR_LITES2_DATA_MATCH_0_0_0_MCHBAR) — Offset 6D10h 711
3.455 B-Unit Lites Group 2 Data Mask Filter LITES2_DATA_MASK (B_CR_LITES2_DATA_MASK_0_0_0_MCHBAR) — Offset 6D14h 712
3.456 B-Unit Lites Group 3 Control (B_CR_LITES3_CTL_0_0_0_MCHBAR) — Offset 6D20h . 712
3.457 B-Unit Lites Group 3 Opcode Match Filter (B_CR_LITES3_OPCODE_MATCH_0_0_0_MCHBAR) — Offset 6D24h 714
3.458 B-Unit Lites Group 3 Agent Match Filter (B_CR_LITES3_AGENT_MATCH_0_0_0_MCHBAR) — Offset 6D28h 715
3.459 B-Unit Lites Group 3 U2C IntData Match Filter (B_CR_LITES3_U2CINTDATA_MATCH_0_0_0_MCHBAR) — Offset 6D2Ch 716
3.460 B-Unit Lites Group 3 Address Match Filter LITES3_ADDR_MATCH (B_CR_LITES3_ADDR_MATCH_0_0_0_MCHBAR) — Offset 6D30h 716
3.461 B-Unit Lites Group 3 Address Mask Filter LITES3_ADDR_MASK (B_CR_LITES3_ADDR_MASK_0_0_0_MCHBAR) — Offset 6D38h 716
3.462 B-Unit Lites Group 3 Data Match Filter LITES3_DATA_MATCH (B_CR_LITES3_DATA_MATCH_0_0_0_MCHBAR) — Offset 6D40h 717
3.463 B-Unit Lites Group 3 Data Mask Filter LITES3_DATA_MASK (B_CR_LITES3_DATA_MASK_0_0_0_MCHBAR) — Offset 6D44h 718
3.464 B-Unit Lites and Emon Master Control LITESEMONCTL (B_CR_LITESEMON_CTL_0_0_0_MCHBAR) — Offset 6D48h 718
3.465 B-Unit Arbiter Control BARBCTRL0 (B_CR_BARBCTRL0) — Offset 6D4Ch 719
3.466 B-Unit Arbiter Control BARBCTRL1 (B_CR_BARBCTRL1) — Offset 6D50h 720
3.467 B-Unit Scheduler Control (B_CR_BSCHWT0) — Offset 6D54h 721
3.468 B-Unit Scheduler Control (B_CR_BSCHWT1) — Offset 6D58h 721
3.469 B-Unit Scheduler Control (B_CR_BSCHWT2) — Offset 6D5Ch 722
3.470 B-Unit Scheduler Control (B_CR_BSCHWT3) — Offset 6D60h 723
3.471 B-Unit Flush Control (B_CR_BWFLUSH) — Offset 6D64h 723
3.472 B-Unit Flush Weights (B_CR_BFLWT) — Offset 6D68h 724
3.473 Weighted Scheduling Control of High Priority ISOC and Other Requests (B_CR_BISOCWT) — Offset 6D6Ch 724
3.474 B-Unit Control (B_CR_BCTRL2) — Offset 6D70h 725
3.475 Asset Classification Bits (B_CR_AC_RS0_0_0_0_MCHBAR) — Offset 6D74h 726
3.476 IDI Real-Time Feature Configuration Bits (B_CR_RT_EN_0_0_0_MCHBAR) — Offset 6D78h 727
3.477 B-Unit Control Register 3 (B_CR_BCTRL3) — Offset 6D7Ch 727
3.478 Asymmetric Memory Region 0 (B_CR_ASYM_MEM_REGION0_0_0_0_MCHBAR) — Offset 6E40h 728
3.479 Asymmetric Memory Region 1 (B_CR_ASYM_MEM_REGION1_0_0_0_MCHBAR) — Offset 6E44h 728
3.480 B-Unit Machine Check Mode Low (B_CR_BMCMODE_LOW) — Offset 6E48h 729
3.481 B-Unit Machine Check Mode High (B_CR_BMCMODE_HIGH) — Offset 6E4Ch 729
3.482 Two-Way Asymmetric Memory Region Configuration (B_CR_ASYM_2WAY_MEM_REGION_0_0_0_MCHBAR) — Offset 6E50h 729
3.483 SGX Status (B_CR_SGX_STATUS_0_0_0_MCHBAR) — Offset 6E54h 730
3.484 Thermal Device Mailbox Data0 (P_CR_THERMAL_MAILBOX_DATA0_0_0_0_MCHBAR) — Offset 7000h 731
3.485 Thermal Device Mailbox Data1 (P_CR_THERMAL_MAILBOX_DATA1_0_0_0_MCHBAR) — Offset 7004h 731
3.486 Thermal Device Mailbox Interface (P_CR_THERMAL_MAILBOX_INTERFACE_0_0_0_MCHBAR) — Offset 7008h 731
3.487 Thermal Device IRQ and Lock Configuration (P_CR_THERMAL_DEVICE_IRQ_0_0_0_MCHBAR) — Offset 700Ch 732
3.488 Package Thermal Interrupt Control (P_CR_PKG_THERM_INTERRUPT_0_0_0_MCHBAR) — Offset 7010h 733

3.489 Package Thermal Status (P_CR_PKG_THERM_STATUS_0_0_0_MCHBAR) — Offset 701Ch 734
3.490 LPDDR DRAM Thermal (MR4) Status of Channel 01 (P_CR_MEM_MR4_TEMPERATURE_DEV1_0_0_0_MCHBAR) — Offset 7024h 734
3.491 LPDDR DRAM Thermal (MR4) Status of Channel 10 (P_CR_MEM_MR4_TEMPERATURE_DEV2_0_0_0_MCHBAR) — Offset 7028h 735
3.492 Machine Check Error Source Log (P_CR_MCA_ERROR_SRC_0_0_0_MCHBAR) — Offset 702Ch 735
3.493 DDR Thermal Throttling Control (P_CR_DDR_THERM_THRT_CTRL_0_0_0_MCHBAR) — Offset 7030h 736
3.494 DDR Thermal Interrupt Control (P_CR_DDR_THERM_INTERRUPT_0_0_0_MCHBAR) — Offset 7034h 737
3.495 DDR Thermal Status (P_CR_DDR_THERM_STATUS_0_0_0_MCHBAR) — Offset 7038h 738
3.496 Dram Energy Counter (P_CR_DDR_ENERGY_STATUS_0_0_0_MCHBAR) — Offset 7048h . 739
3.497 DDR RAPL Performance Status (P_CR_DDR_RAPL_PERF_STATUS_0_0_0_MCHBAR) — Offset 704Ch 739
3.498 Package RAPL Performance Status (P_CR_PACKAGE_RAPL_PERF_STATUS_0_0_0_MCHBAR) — Offset 7050h 740
3.499 IA Core Performance / Power Priority Control (P_CR_PRIMARY_PLANE_TURBO_PLCY_0_0_0_MCHBAR) — Offset 7054h 741
3.500 Graphics Performance / Power Priority Control (P_CR_SECONDARY_PLANE_TURBO_PLCY_0_0_0_MCHBAR) — Offset 7058h 741
3.501 IA Energy Counter (P_CR_PRIMARY_PLANE_ENERGY_STATUS_0_0_0_MCHBAR) — Offset 705Ch 742
3.502 Graphics Energy Counter (P_CR_SECONDARY_PLANE_ENERGY_STATUS_0_0_0_MCHBAR) — Offset 7060h 742
3.503 PACKAGE_POWER_SKU_UNIT (P_CR_PACKAGE_POWER_SKU_UNIT_0_0_0_MCHBAR) — Offset 7068h 743
3.504 SOC Energy Counter (P_CR_PACKAGE_ENERGY_STATUS_0_0_0_MCHBAR) — Offset 706Ch 743
3.505 GT_PERF_STATUS (P_CR_GT_PERF_STATUS_0_0_0_MCHBAR) — Offset 7070h..... 744
3.506 Temperature Reference and Control (P_CR_TEMPERATURE_TARGET_0_0_0_MCHBAR) — Offset 7074h 744
3.507 BIOS Reset Completion (P_CR_BIOS_RESET_CPL_0_0_0_MCHBAR) — Offset 7078h 745
3.508 BIOS_MAILBOX_DATA (P_CR_BIOS_MAILBOX_DATA_0_0_0_MCHBAR) — Offset 7080h . 746
3.509 BIOS_MAILBOX_INTERFACE (P_CR_BIOS_MAILBOX_INTERFACE_0_0_0_MCHBAR) — Offset 7084h 747
3.510 CORE_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_CORE_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR) — Offset 7088h 747
3.511 GRAPHICS_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_GRAPHICS_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR) — Offset 708Ch 748
3.512 SYSTEM_AGENT_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_SYSTEM_AGENT_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR) — Offset 7090h 748
3.513 Memory Frequency Status (P_CR_FAR_MEMORY_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR) — Offset 7094h 749
3.514 Package Power SKU and RAPL Power Control Capabilities (P_CR_PACKAGE_POWER_SKU_0_0_0_MCHBAR) — Offset 70A0h 750
3.515 Package RAPL Power Limit (P_CR_PACKAGE_RAPL_LIMIT_0_0_0_MCHBAR) — Offset 70A8h 750
3.516 IA_PERF_LIMIT_REASONS (P_CR_IA_PERF_LIMIT_REASONS_0_0_0_MCHBAR) — Offset 70B0h 752
3.517 IA Core C0 Residency Counter (P_CR_TELEM_IA_C0_RESIDENCY_0_0_0_MCHBAR) — Offset 70C0h 754
3.518 Graphics C0 Residency Counter (P_CR_TELEM_GT_C0_RESIDENCY_0_0_0_MCHBAR) —

Offset 70C4h 754
3.519 I-unit Processing System C0 Residency Counter (P_CR_TELEM_IUNIT_C0_RESIDENCY_0_0_0_MCHBAR) — Offset 70C8h 755
3.520 TELEM_IA_FREQ_ACCUMULATOR (P_CR_TELEM_IA_FREQ_ACCUMULATOR_0_0_0_MCHBAR) — Offset 70CCh 755
3.521 Graphics C0 Residency Counter (P_CR_TELEM_GT_FREQ_ACCUMULATOR_0_0_0_MCHBAR) — Offset 70D0h 755
3.522 I-unit Processing System C0 Residency Counter (P_CR_TELEM_IUNIT_FREQ_ACCUMULATOR_0_0_0_MCHBAR) — Offset 70D4h 756
3.523 Memory Active Residency (P_CR_TELEM_FAR_MEMORY_ACTIVE_0_0_0_MCHBAR) — Offset 70E8h 756
3.524 Package Temperatures (P_CR_PACKAGE_TEMPERATURES_0_0_0_MCHBAR) — Offset 70F4h 757
3.525 Package Thermal Limit Control (P_CR_THERMAL_LIMIT_CONTROL_0_0_0_MCHBAR) — Offset 7104h 757
3.526 Memory Subsystem Frequency Capabilities (P_CR_MEMSS_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR) — Offset 7108h 758
3.527 Memory Controller (MC) BIOS Reset Request and Status (P_CR_MC_BIOS_REQ_0_0_0_MCHBAR) — Offset 7114h 758
3.528 P_CR_MEMSS_FREQUENCY_CAPABILITIES1_0_0_0_MCHBAR — Offset 7118h 761
3.529 PP1_C0_CORE_CLOCK_0_0_0_MCHBAR (P_CR_PP1_C0_CORE_CLOCK_0_0_0_MCHBAR) — Offset 7160h 761
3.530 Core Exists Vector (P_CR_CORE_EXISTS_VECTOR_0_0_0_MCHBAR) — Offset 7164h 762
3.531 Software Core Disable Mask (P_CR_CORE_DISABLE_MASK_0_0_0_MCHBAR) — Offset 7168h 762
3.532 PL3 and PL4 Control (P_CR_PL3_CONTROL_0_0_0_MCHBAR) — Offset 71F0h 763
3.533 Graphics Superqueue Active Clocks (P_CR_PP1_ANY_THREAD_ACTIVITY_0_0_0_MCHBAR) — Offset 7244h 764
3.534 LPDDR DRAM Thermal (MR4) Status of Channel 00 (P_CR_MEM_MR4_TEMPERATURE_DEV3_0_0_0_MCHBAR) — Offset 7248h 765
3.535 LPDDR DRAM Thermal (MR4) Status of Channel 11 (P_CR_MEM_MR4_TEMPERATURE_DEV4_0_0_0_MCHBAR) — Offset 724Ch 765
3.536 Interrupt Redirection Control (INTR_REDIR_CTL_MCHBAR) — Offset 7800h 765
3.537 X2B_BARB_CTL1 (X2B_BARB_CTL1_MCHBAR) — Offset 7804h 766
3.538 Clock Gating Control (CLKGATE_CTL_MCHBAR) — Offset 7808h 767
3.539 Miscellaneous Controls (MISC_CTL_MCHBAR) — Offset 780Ch 769
3.540 Control (CTL_MCHBAR) — Offset 7810h 770
3.541 PSMI Control (PSMI_CTL_MCHBAR) — Offset 7814h 772
3.542 Miscellaneous T2A selector (T2A_SELECTOR_MISC_MCHBAR) — Offset 7818h 773
3.543 VC Read Ordering CFG (VC_READ_ORDERING_CFG_MCHBAR) — Offset 781Ch 774
3.544 VC Write Ordering CFG (VC_WRITE_ORDERING_CFG_MCHBAR) — Offset 7820h 775
3.545 IDI0 C2U Credit Control (IDI0_C2U_CREDIT_CTRL_MCHBAR) — Offset 7824h 777
3.546 IDI1 C2U Credit Control (IDI1_C2U_CREDIT_CTRL_MCHBAR) — Offset 7828h 777
3.547 IDI2 C2U Credit Control (IDI2_C2U_CREDIT_CTRL_MCHBAR) — Offset 782Ch 778
3.548 IDI3 C2U Credit Control (IDI3_C2U_CREDIT_CTRL_MCHBAR) — Offset 7830h 779
3.549 IDI4 C2U Credit Control (IDI4_C2U_CREDIT_CTRL_MCHBAR) — Offset 7834h 780
3.550 IDI5 C2U Credit Control (IDI5_C2U_CREDIT_CTRL_MCHBAR) — Offset 7838h 780
3.551 IDI6 C2U Credit Control (IDI6_C2U_CREDIT_CTRL_MCHBAR) — Offset 783Ch 781
3.552 IDI7 C2U Credit Control (IDI7_C2U_CREDIT_CTRL_MCHBAR) — Offset 7840h 782
3.553 PII2 A2T Credit Control (PII2_A2T_CREDIT_CTRL_MCHBAR) — Offset 7844h 782
3.554 BIOSWR Control Policy (T_CR_BIOSWR_CP_0_0_0_MCHBAR) — Offset 7848h 783
3.555 BIOSWR Read Access Control (T_CR_BIOSWR_RAC_0_0_0_MCHBAR) — Offset 7850h... 783
3.556 BIOSWR Write Access Control (T_CR_BIOSWR_WAC_0_0_0_MCHBAR) — Offset 7858h . 784
3.557 Tunit Pcode/Ucode Write, All Read Control Policy Register (T_CR_P_U_CODEWR_ALL-

RD_CP_0_0_0_MCHBAR) — Offset 7860h 784
3.558 TUnit Pcode/Ucode Read Access Control (T_CR_P_U_CODEWR_ALLRD_RAC_0_0_0_MCHBAR) — Offset 7868h 784
3.559 TUnit Pcode/Ucode Write Access Control (T_CR_P_U_CODEWR_ALLRD_WAC_0_0_0_MCHBAR) — Offset 7870h 785
3.560 PMTA_WEIGHTED_ARB_CONFIG (PMTA_WEIGHTED_ARB_CONFIG)—Offset 4C04h .. 785
3.561 PMTA_QOS_CONFIG0 (PMTA_QOS_CONFIG0)—Offset 4C08h 786
3.562 PMTA_QOS_CONFIG1 (PMTA_QOS_CONFIG1)—Offset 4C10h 787
3.563 PMTA_MEE_BIOS_CONFIG (PMTA_MEE_BIOS_CONFIG)—Offset 4C18h..... 788

4 Processor Graphics Registers 792

4.1 B-Unit Copy of PCICMD for IGD (B_CR_PCICMD_0_2_0_PCI) — Offset 4h 792
4.2 B-Unit Copy of GTTMMADR (B_CR_GTTMMADR_LO_0_2_0_PCI) — Offset 10h..... 793
4.3 B-Unit Copy of GTTMMADR (B_CR_GTTMMADR_HI_0_2_0_PCI) — Offset 14h 793
4.4 B-Unit Copy of GMADR (B_CR_GMADR_LO_0_2_0_PCI) — Offset 18h 793
4.5 B-Unit Copy of GMADR (B_CR_GMADR_HI_0_2_0_PCI) — Offset 1Ch..... 794
4.6 B-Unit Copy of the IOBAR (B_CR_IOBAR_0_2_0_PCI) — Offset 20h 795
4.7 B-Unit Copy of Device 2 Control Register (B_CR_DEV2CTL_0_2_0_PCI) — Offset 58h..... 795
4.8 B-Unit Copy of MSAC (B_CR_MSAC_0_2_0_PCI) — Offset 62h 795
4.9 B-Unit Copy of Device 2 Control Register (B_CR_DEVICECTL_0_2_0_PCI) — Offset 78h.. 796
4.10 B-Unit Copy of PMCS for IGD (B_CR_PMCS_0_2_0_PCI) — Offset D4h 797

5 System Agent Address Map Registers 799

5.1 Device ID and Vendor ID Register (P_CR_DEVICE_ID_VENDOR_ID_0_0_1_PCI) — Offset 0h 799
5.2 PCI_STATUS_COMMAND_0_0_1_PCI (B_CR_PCI_STATUS_COMMAND_0_0_1_PCI) — Offset 4h..... 799
5.3 PCI Revision ID and PCI Class Code Register (P_CR_REVISION_ID_-CLASS_CODE_0_0_1_PCI) — Offset 8h 801
5.4 Master Latency Timer and Header Type Register (P_CR_MASTER_LATENCY_-TIME_0_0_1_PCI) — Offset Ch..... 801
5.5 Thermal Management Base Address Register (B_CR_TMBAR_LO_0_0_1_PCI) — Offset 10h 802
5.6 Thermal Management Base Address Register (B_CR_TMBAR_HI_0_0_1_PCI) — Offset 14h 802
5.7 PCI Subsystem Vendor ID and PCI Subsystem ID (P_CR_SVID_SID_0_0_1_PCI) — Offset 2Ch 803
5.8 CAPPTR_0_0_1_PCI (P_CR_CAPPTR_0_0_1_PCI) — Offset 34h 803
5.9 Interrupt and Latency Configuration (P_CR_INTR_LAT_0_0_1_PCI) — Offset 3Ch 803
5.10 Device Enable Register (P_CR_DEVEN_0_0_1_PCI) — Offset 54h..... 804
5.11 SCISTS_0_0_1_PCI (P_CR_SCISTS_0_0_1_PCI) — Offset 88h 805
5.12 SCI Command (P_CR_SCICMD_0_0_1_PCI) — Offset CCh 805
5.13 Power Management Capabilities (P_CR_PMCAPID_0_0_1_PCI) — Offset D0h..... 806
5.14 Power Management Control and Status (P_CR_PMCS_0_0_1_PCI) — Offset D4h..... 807
5.15 Interrupt Status (P_CR_INTSTS_0_0_1_PCI) — Offset DCh..... 808
5.16 Capability ID0 Capability Control (P_CR_CAPID0_CAPCTRL0_0_0_1_PCI) — Offset E0h .. 808
5.17 Capability ID0 A (P_CR_CAPID0_A_0_0_1_PCI) — Offset E4h 809
5.18 Capability ID0 B (P_CR_CAPID0_B_0_0_1_PCI) — Offset E8h 810
5.19 B-Unit Copy of Default VTd BAR PMEN (B_CR_PMEN_REG_0_0_0_DEFVTDBAR)—Offset 64h 811
5.20 B-Unit Copy of Default VTd BAR PLM Base Register (B_CR_PLM-BASE_REG_0_0_0_DEFVTDBAR)—Offset 68h 811
5.21 B-Unit Copy of Default VTd BAR PLM Limit Register (B_CR_PLMLIM-IT_REG_0_0_0_DEFVTDBAR)—Offset 6Ch 812

5.22 B-Unit Copy of Default VTd BAR PHM Base Register (B_CR_PHM-BASE_REG_0_0_0_DEFVTDBAR)—Offset 70h 813
5.23 B-Unit Copy of Default VTd BAR PHM Limit Register (B_CR_PHMLIM-IT_REG_0_0_0_DEFVTDBAR)—Offset 78h 813

6 Gaussian Mixture Model (GMM) Registers 815

6.1 ID — Offset 0h 815
6.2 Device Control (DCTRL) — Offset 4h 815
6.3 Device Status (DSTS) — Offset 6h 816
6.4 RID: Revision ID DLCO: Class Code (RID_DLCO) — Offset 8h 817
6.5 Cache Line Size (CLS) — Offset Ch 817
6.6 Header Type (HTYPE) — Offset Eh 818
6.7 Built-in Self Test (BIST) — Offset Fh 818
6.8 GNA Base Address Low (GNABAL) — Offset 10h 818
6.9 GNA Base Address High (GNABAH) — Offset 14h 819
6.10 Sub System Vendor Identifiers (SSVI) — Offset 2Ch 819
6.11 Sub System Identifiers (SSI) — Offset 2Eh 820
6.12 Capabilities Pointers (CAPP) — Offset 34h 820
6.13 Interrupt Line (INTL) — Offset 3Ch 820
6.14 Interrupt Pin Register (INTP) — Offset 3Dh 821
6.15 Min Grant And Min Latency Register (MINGNTLAT) — Offset 3Eh 821
6.16 Override Configuration Control (OVRCFGCTL) — Offset 40h 821
6.17 Message Signaled Interrupt Capability ID (MSICAPID) — Offset 90h 823
6.18 Message Signaled Interrupt Message Control (MC) — Offset 92h 823
6.19 Message Signaled Interrupt Message Address (MA) — Offset 94h 824
6.20 Message Signaled Interrupt Message Data (MD) — Offset 98h 824
6.21 D0i3 Capability ID (D0I3CAPID) — Offset A0h 824
6.22 D0i3 Capability (D0I3CAP) — Offset A2h 825
6.23 D0i3 Vendor Extended Capability Register (D0I3VSEC) — Offset A4h 825
6.24 D0i3 SW LTR Pointer Register (D0I3SWLTRPTR) — Offset A8h 826
6.25 D0i3 DevIdle Pointer Register (D0I3DEVIDLEPTR) — Offset ACh 826
6.26 D0i3 DevIdle Power On Latency (D0I3DEVIDLEPOL) — Offset B0h 826
6.27 D0i3 Power Control Enables Register (PCE) — Offset B2h 827
6.28 Power Management Capability ID (PMCAPID) — Offset DCh 827
6.29 Power Management Capability (PMCAP) — Offset DEh 828
6.30 Power Management Control Status (PMCS) — Offset E0h 828
6.31 FLR Capability ID (FLRCAPID) — Offset F0h 829
6.32 FLR Capability Length And Version (FLRMISC) — Offset F2h 829
6.33 FLR Control Register (FLRCTL) — Offset F4h 830
6.34 FLR Status Register (FLRSTS) — Offset F5h 830
6.35 GMM Status Register (GNASTS)—Offset 80h 830
6.36 GMM Control Register (GNACTRL)—Offset 84h 832
6.37 GMM Management And Control Register (GNAMCTL)—Offset 88h 834
6.38 GMM Performance Total Cycle (GNAPTC)—Offset 8Ch 834
6.39 GMM Performance Stall Cycles (GNAPSC)—Offset 90h 835
6.40 GMM Internal State Index (GNAISI)—Offset 94h 835
6.41 GMM Internal State Value Low (GNAPISVL)—Offset 98h 836
6.42 GMM Internal State Value High (GNAPISVH)—Offset 9Ch 836
6.43 GMM Break Point Setup Low (GNABPL)—Offset A0h 836
6.44 GMM Break Point Setup High (GNABPH)—Offset A4h 837
6.45 GMM Internal State Value (GNAD0i3C)—Offset A8h 837
6.46 GNA Descriptor Base Address Register (GNADESBASE)—Offset B0h 838
6.47 GNA Information Register (GNAIBUFFS)—Offset B4h 839
6.48 GNASAI1L (GNASAI1L)—Offset 100h 839
6.49 GNASAI1H (GNASAI1H)—Offset 104h 839

6.50 GNASAI2 (GNASAI2L)—Offset 108h 840
6.51 GNASAI2 (GNASAI2H)—Offset 10Ch 840
6.52 GNASAI2 (GNASAIV)—Offset 110h 840

7 SMBus Registers 841

7.1 Vendor ID (VID) — Offset 0h 841
7.2 Device ID (DID) — Offset 2h 841
7.3 Command (CMD) — Offset 4h 841
7.4 Device Status (DS) — Offset 6h 842
7.5 Revision ID (RID) — Offset 8h 843
7.6 Programming Interface (PI) — Offset 9h 843
7.7 Sub Class Code (SCC) — Offset Ah 844
7.8 Base Class Code (BCC) — Offset Bh 844
7.9 SMBus Memory Base Address_31_0 (SMBMBAR_31_0) — Offset 10h 844
7.10 SMBus Memory Base Address_63_32 (SMBMBAR_63_32) — Offset 14h 845
7.11 SMB Base Address (SBA) — Offset 20h 845
7.12 SVID — Offset 2Ch 845
7.13 SID — Offset 2Eh 846
7.14 Interrupt Line (INTLN) — Offset 3Ch 846
7.15 Interrupt Pin (INTPN) — Offset 3Dh 846
7.16 Host Configuration (HCFG) — Offset 40h 847
7.17 B-Unit Copy of the TCO Base Address Register for Legacy SMBUS (B_CR_TCOBASE_SMBUS) — Offset 50h 847
7.18 B-Unit Copy of the TCO Control Register for Legacy SMBUS (B_CR_TCOCTL_SMBUS) — Offset 54h 848
7.19 Manufacturer's ID (MANID) — Offset F8h 848
7.20 Host Status Register Address (HSTS)—Offset 0h 849
7.21 Host Control Register (HCTL)—Offset 2h 850
7.22 Host Command Register (HCMD)—Offset 3h 853
7.23 Transmit Slave Address Register (TSA)—Offset 4h 853
7.24 Data 0 Register (HD0)—Offset 5h 853
7.25 Data 1 Register (HD1)—Offset 6h 854
7.26 Host Block Data (HBD)—Offset 7h 854
7.27 Packet Error Check Data Register (PEC)—Offset 8h 855
7.28 Receive Slave Address Register (RSA)—Offset 9h 856
7.29 Slave Data Register (SD)—Offset Ah 856
7.30 Auxiliary Status (AUXS)—Offset Ch 857
7.31 Auxiliary Control (AUXC)—Offset Dh 857
7.32 SMLINK_PIN_CTL Register (SMLC)—Offset Eh 858
7.33 SMBUS_PIN_CTL Register (SMBC)—Offset Fh 858
7.34 Slave Status Register (SSTS)—Offset 10h 859
7.35 Slave Command Register (SCMD)—Offset 11h 860
7.36 Notify Device Address Register (NDA)—Offset 14h 861
7.37 Notify Data Low Byte Register (NDLB)—Offset 16h 861
7.38 Notify Data High Byte Register (NDHB)—Offset 17h 861
7.39 TCO_RLD Register (TRLD)—Offset 0h 862
7.40 TCO_DAT_IN Register (TDI)—Offset 2h 862
7.41 TCO_DAT_OUT Register (TDO)—Offset 3h 862
7.42 TCO1_STS Register (TSTS1)—Offset 4h 863
7.43 TCO2_STS Register (TSTS2)—Offset 6h 865
7.44 TCO1_CNT Register (TCTL1)—Offset 8h 866
7.45 TCO2_CNT Register (TCTL2)—Offset Ah 867
7.46 TCO_MESSAGE1 and TCO_MESSAGE2 (TMSG)—Offset Ch 868
7.47 TCO_WDSTATUS Register (TWDS)—Offset Eh 868
7.48 Reserved_1 (RSV_1)—Offset Fh 869

7.49 LEGACY_ELIM Register (LE)—Offset 10h 869
7.50 Reserved_2 (RSV_2)—Offset 11h 869
7.51 TCO_TMR Register (TTMR)—Offset 12h 870
7.52 Reserved_3 (RSV_3)—Offset 14h 870
7.53 Reserved_4 (RSV_4)—Offset 1Ch 870

8 LPC Registers 872

8.1 ID — Offset 0h 872
8.2 Device Command (CMD) — Offset 4h 872
8.3 Status (STS) — Offset 6h 873
8.4 Revision ID (RID) — Offset 8h 874
8.5 Class Code (CC) — Offset 9h 874
8.6 Primary Latency Timer (PLT) — Offset Dh 874
8.7 Header Type (HTYPE) — Offset Eh 875
8.8 Sub System Identifiers (SS) — Offset 2Ch 875
8.9 Capability List Pointer (CAPP) — Offset 34h 875
8.10 Serial IRQ Control (SCNT) — Offset 64h 876
8.11 B-Unit Copy of the I/O Decode Ranges Register for LPC (B_CR_IOD_IOE_LPC) — Offset 80h 876
8.12 I/O Enables (IOE) — Offset 82h 878
8.13 B_LGIR1_LPC (B_CR_LGIR1_LPC) — Offset 84h 878
8.14 B_LGIR2_LPC (B_CR_LGIR2_LPC) — Offset 88h 879
8.15 B_LGIR3_LPC (B_CR_LGIR3_LPC) — Offset 8Ch 879
8.16 B_LGIR4_LPC (B_CR_LGIR4_LPC) — Offset 90h 880
8.17 USB Legacy Keyboard/Mouse Control (ULKMC) — Offset 94h 881
8.18 B-Unit Copy of the LPC Generic Memory Range Register for LPC (B_CR_LGMR_LPC) — Offset 98h 882
8.19 Reserved (RSVD) — Offset 9Ch 882
8.20 B-Unit Copy of the eSPI CS1# Generic I/O Range #1 (B_CR_PCCS1GIR1_ESPI) — Offset A4h 882
8.21 B-Unit Copy of the eSPI CS1# Generic Memory Range #1 Register (B_CR_PCCS1GMR1_ESPI) — Offset A8h 883
8.22 FWH ID Select #1 (FS1) — Offset D0h 883
8.23 FWH ID Select #2 (FS2) — Offset D4h 884
8.24 B-Unit Copy of the BIOS Decode Enable Register for LPC (B_CR_BDE_LPC) — Offset D8h 884
8.25 BIOS Control (BC) — Offset DCh 885
8.26 RBR (RBR)—Offset 0h 886
8.27 THR (THR)—Offset 0h 886
8.28 DLL (DLL)—Offset 0h 887
8.29 IER (IER)—Offset 4h 887
8.30 DLH (DLH)—Offset 4h 888
8.31 IIR (IIR)—Offset 8h 888
8.32 FCR (FCR)—Offset 8h 888
8.33 LCR (LCR)—Offset Ch 889
8.34 MCR (MCR)—Offset 10h 890
8.35 LSR (LSR)—Offset 14h 892
8.36 MSR (MSR)—Offset 18h 894
8.37 SCR (SCR)—Offset 1Ch 895
8.38 SRBR_STHR0 (SRBR_STHR0)—Offset 30h 895
8.39 SRBR_STHR1 (SRBR_STHR1)—Offset 34h 896
8.40 SRBR_STHR2 (SRBR_STHR2)—Offset 38h 896
8.41 SRBR_STHR3 (SRBR_STHR3)—Offset 3Ch 897
8.42 SRBR_STHR4 (SRBR_STHR4)—Offset 40h 898
8.43 SRBR_STHR5 (SRBR_STHR5)—Offset 44h 898
8.44 SRBR_STHR6 (SRBR_STHR6)—Offset 48h 899

8.45 SRBR_STHR7 (SRBR_STHR7)—Offset 4Ch 900
8.46 SRBR_STHR8 (SRBR_STHR8)—Offset 50h 900
8.47 SRBR_STHR9 (SRBR_STHR9)—Offset 54h 901
8.48 SRBR_STHR10 (SRBR_STHR10)—Offset 58h 902
8.49 SRBR_STHR11 (SRBR_STHR11)—Offset 5Ch 902
8.50 SRBR_STHR12 (SRBR_STHR12)—Offset 60h 903
8.51 SRBR_STHR13 (SRBR_STHR13)—Offset 64h 904
8.52 SRBR_STHR14 (SRBR_STHR14)—Offset 68h 904
8.53 SRBR_STHR15 (SRBR_STHR15)—Offset 6Ch 905
8.54 FAR (FAR)—Offset 70h 906
8.55 TFR (TFR)—Offset 74h 906
8.56 RFW (RFW)—Offset 78h 907
8.57 USR (USR)—Offset 7Ch 907
8.58 TFL (TFL)—Offset 80h 908
8.59 RFL (RFL)—Offset 84h 908
8.60 SRR (SRR)—Offset 88h 909
8.61 SRTS (SRTS)—Offset 8Ch 909
8.62 SBCR (SBCR)—Offset 90h 910
8.63 SDMAM (SDMAM)—Offset 94h 911
8.64 SFE (SFE)—Offset 98h 911
8.65 SRT (SRT)—Offset 9Ch 911
8.66 STET (STET)—Offset A0h 912
8.67 HTX (HTX)—Offset A4h 912
8.68 DMASA (DMASA)—Offset A8h 913
8.69 CPR (CPR)—Offset F4h 913
8.70 UCV (UCV)—Offset F8h 914
8.71 CTR (CTR)—Offset FCh 915
8.72 RID - Bridge revision ID Register (RID)—Offset 30h 915
8.73 GEN_REGRW1 - General Purpose register (GEN_REGRW1)—Offset 600h 915
8.74 GEN_REGRW2 - General Purpose register (GEN_REGRW2)—Offset 604h 916
8.75 GEN_REGRW3 - General Purpose register (GEN_REGRW3)—Offset 608h 916
8.76 GEN_REGRW4 - General Purpose register (GEN_REGRW4)—Offset 60Ch 916
8.77 Identifiers (ID)—Offset 0h 917
8.78 Device Command (CMD)—Offset 4h 917
8.79 Status (STS)—Offset 6h 918
8.80 Revision ID (RID)—Offset 8h 919
8.81 Class Code (CC)—Offset 9h 919
8.82 Primary Latency Timer (PLT)—Offset Dh 920
8.83 Header Type (HTYPE)—Offset Eh 920
8.84 Sub System Identifiers (SS)—Offset 2Ch 921
8.85 Capability List Pointer (CAPP)—Offset 34h 921
8.86 Serial IRQ Control (SCNT)—Offset 64h 921
8.87 I/O Decode Ranges (IOD)—Offset 80h 922
8.88 I/O Enables (IOE)—Offset 82h 923
8.89 LPC Generic I/O Range #1 (LGIR1)—Offset 84h 924
8.90 LPC Generic I/O Range #2 (LGIR2)—Offset 88h 925
8.91 LPC Generic I/O Range #3 (LGIR3)—Offset 8Ch 925
8.92 LPC Generic I/O Range #4 (LGIR4)—Offset 90h 926
8.93 USB Legacy Keyboard/Mouse Control (ULKMC)—Offset 94h 926
8.94 LPC Generic Memory Range (LGMR)—Offset 98h 928
8.95 Reserved (RSVD)—Offset 9Ch 928
8.96 FWH ID Select #1 (FS1)—Offset D0h 929
8.97 FWH ID Select #2 (FS2)—Offset D4h 929
8.98 BIOS Decode Enable (BDE)—Offset D8h 930
8.99 BIOS Control (BC)—Offset DCh 931

8.100 PCI Clock Control (PCCTL)—Offset E0h 933
8.101 Manufacturer's ID (MANID)—Offset F8h 935

9 eSPI Registers 936

9.1 Identifiers (ESPI_DID_VID) — Offset 0h 936
9.2 Device Status and Command (ESPI_STS_CMD) — Offset 4h 936
9.3 Class Code and Revision ID (ESPI_CC_RID) — Offset 8h 937
9.4 Peripheral Channel Header Type and Primary Latency (ESPI_PCHT_PCPLT) — Offset Ch . 938
9.5 CSXE BAR0 MMIO (CSXE_ESPI_MBAR) — Offset 10h 938
9.6 Sub System Identifiers (ESPI_SS) — Offset 2Ch 939
9.7 Capability List Pointer (ESPI_CAPP) — Offset 34h 939
9.8 Capabilities List Pointer (CSXE_ESPI_CAPP) — Offset 40h 939
9.9 MSI Message Control, Next Pointer and Capability ID (CSXE_ESPI_MSIMC_MSINP_MSI-CID) — Offset 44h 940
9.10 MSI Message Address (CSXE_ESPI_MSIMA) — Offset 48h 940
9.11 MSI Message Data (CSXE_ESPI_MSIMD) — Offset 4Ch 941
9.12 PCI Power Management Capability (CSXE_ESPI_PMCAP_PMNP_PMCID) — Offset 50h 941
9.13 PCI Power Management Control and Status (CSXE_ESPI_PMD_PMCSRBSE_PMCSR) — Offset 54h 942
9.14 Peripheral Channel Serial IRQ Control (ESPI_PCSERIRQC) — Offset 64h 943
9.15 I/O Decode Ranges and I/O Enables (ESPI_IOD_IOE) — Offset 80h 943
9.16 LPC Generic I/O Range #1 (ESPI_LGIR1) — Offset 84h 944
9.17 LPC Generic I/O Range #2 (ESPI_LGIR2) — Offset 88h 945
9.18 LPC Generic I/O Range #3 (ESPI_LGIR3) — Offset 8Ch 945
9.19 LPC Generic I/O Range #4 (ESPI_LGIR4) — Offset 90h 946
9.20 USB Legacy Keyboard/Mouse Control (ESPI_ULKMC) — Offset 94h 946
9.21 LPC Generic Memory Range (ESPI_LGMR) — Offset 98h 947
9.22 eSPI CS1# I/O Routing Enables (ESPI_CS1IORE) — Offset A0h 948
9.23 eSPI CS1# Generic I/O Range #1 (ESPI_CS1GIR1) — Offset A4h 949
9.24 eSPI CS1# Generic Memory Range #1 (ESPI_CS1GMR1) — Offset A8h 949
9.25 FWH ID Select #1 (ESPI_PCFS1) — Offset D0h 950
9.26 FWH ID Select #2 (ESPI_PCFS2) — Offset D4h 950
9.27 BIOS Decode Enable (ESPI_BDE) — Offset D8h 950
9.28 BIOS Control (ESPI_BC) — Offset DCh 951
9.29 Manufacturer's ID (ESPI_MANID) — Offset F8h 953

10 PMC Registers 954

10.1 DEVICEVENDORID - Device ID and Vendor ID Register (DEVVENDID) — Offset 0h ... 954
10.2 STATUSCOMMAND — Offset 4h 954
10.3 REVCLASSCODE — Offset 8h 955
10.4 CLLATHEADERBIST — Offset Ch 955
10.5 BAR — Offset 10h 956
10.6 BAR -Base Address Register High (BAR_HIGH) — Offset 14h 956
10.7 BAR1 — Offset 18h 957
10.8 BAR1 -Base Address Register1 High (BAR1_HIGH) — Offset 1Ch 957
10.9 BAR2 — Offset 20h 958
10.10 SUBSYSTEMID — Offset 2Ch 958
10.11 EXPANSION ROM base address (EXPANSION_ROM_BASEADDR) — Offset 30h 958
10.12 CAPABILITYPTR — Offset 34h 959
10.13 INTERRUPTREG — Offset 3Ch 959
10.14 POWERCAPID — Offset 80h 959
10.15 PMECTRLSTATUS — Offset 84h 960
10.16 PCI DEVICE IDLE CAPABILITY RECORD (PCIDEVIDLE_CAP_RECORD) — Offset 90h ... 960
10.17 DEVID VENDOR SPECIFIC REG (DEVID_VEND_SPECIFIC_REG) — Offset 94h 961
10.18 D0I3_CONTROL_SW_LTR_MMIO_REG — Offset 98h 961

10.19 DEVICE_IDLE_POINTER_REG — Offset 9Ch 962
10.20 D0I3_MAX_POW_LAT_PG_CONFIG — Offset A0h 962
10.21 GEN_REGRW1 - General Purpose Read Write Register1 (GEN_REGRW1) — Offset B0h 963
10.22 GEN_REGRW2 — Offset B4h 963
10.23 GEN_REGRW3 — Offset B8h 963
10.24 GEN_REGRW4 — Offset BCh 964
10.25 GEN_INPUT_REG — Offset C0h 964
10.26 Manufacturers ID (MANID) — Offset F8h 964
10.27 THREAD_STATUS (P_CR_GT_THREAD_STATUS_0_2_0_GTTMMADR)—Offset 805Ch . 965
10.28 CORE_STATUS (P_CR_GT_CORE_STATUS_0_2_0_GTTMMADR)—Offset 8060h 966
10.29 GT_SLICE_INFO (P_CR_GT_SLICE_INFO_0_2_0_GTTMMADR)—Offset 8064h 967
10.30 GT Hardware P-state Control Request (P_CR_GT_HWP_REQ_0_2_0_GTTMMADR)—Offset 8068h 968
10.31 GT_MEM_BOUND_COUNTER_0_2_0_GTTMMADR (P_CR_GT_MEM_BOUND_COUNTER_0_2_0_GTTMMADR)—Offset 8070h 969
10.32 GT_SQ_OCCUPANCY_0_2_0_GTTMMADR (P_CR_GT_SQ_OCCUPANCY_0_2_0_GTTM-MADR)—Offset 8074h 969
10.33 GT_RW_DRAM_0_2_0_GTTMMADR (P_CR_GT_RW_DRAM_0_2_0_GTTMMADR)—Offset 8078h 969
10.34 GT_P_REQ (P_CR_GT_THREAD_P_REQ_0_2_0_GTTMMADR)—Offset 807Ch 970
10.35 GT_ARAT_TTT (P_CR_GT_ARAT_TTT_0_2_0_GTTMMADR)—Offset 8080h 970
10.36 GTC6_PREWAKE_TIMER_0_2_0_GTTMMADR (P_CR_GTC6_PREWAKE_TIM-ER_0_2_0_GTTMMADR)—Offset 8088h 971
10.37 GT_DISP_PWRON_0_2_0_GTTMMADR (P_CR_GT_DISP_PWRON_0_2_0_GTTMMADR)—Offset 8090h 971
10.38 GT_GFX_RC6_0_2_0_GTTMMADR (P_CR_GT_GFX_RC6_0_2_0_GTTMMADR)—Offset 8108h 972
10.39 GTDRIVER_MAILBOX_INTERFACE_0_2_0_GTTMMADR (P_CR_GTDRIVER_MAILBOX_IN-TERFACE_0_2_0_GTTMMADR)—Offset 8124h 972
10.40 GTDRIVER_MAILBOX_DATA_LOW_0_2_0_GTTMMADR (P_CR_GTDRIVER_MAILBOX_-DATA_LOW_0_2_0_GTTMMADR)—Offset 8128h 973
10.41 GTDRIVER_MAILBOX_DATA_HIGH_0_2_0_GTTMMADR (P_CR_GTDRIVER_MAILBOX_-DATA_HIGH_0_2_0_GTTMMADR)—Offset 812Ch 973
10.42 P24C_PCODE_MAILBOX_INTERFACE_0_2_0_GTTMMADR (P_CR_P24C_PCODE_MAIL-BOX_INTERFACE_0_2_0_GTTMMADR)—Offset 8130h 974
10.43 P24C_PCODE_MAILBOX_DATA_0_2_0_GTTMMADR (P_CR_P24C_PCODE_MAILBOX_-DATA_0_2_0_GTTMMADR)—Offset 8134h 975
10.44 PCODE_P24C_MAILBOX_INTERFACE_0_2_0_GTTMMADR (P_CR_PCODE_P24C_MAIL-BOX_INTERFACE_0_2_0_GTTMMADR)—Offset 8138h 975
10.45 PCODE_P24C_MAILBOX_DATA_0_2_0_GTTMMADR (P_CR_PCODE_P24C_MAILBOX_-DATA_0_2_0_GTTMMADR)—Offset 813Ch 976
10.46 GT_PM_CONFIG_0_2_0_GTTMMADR (P_CR_GT_PM_CONFIG_0_2_0_GTTMMADR)—Off-set 8140h 976
10.47 Graphics Interrupt Response Latency Tolerance (P_CR_GRAPHICS_INTERRUPT_RESPON-SE_TIME_0_2_0_GTTMMADR)—Offset 8150h 977
10.48 GTDRIVER_P2G_EVENTS_0_2_0_GTTMMADR (P_CR_GTDRIV-ER_P2G_EVENTS_0_2_0_GTTMMADR)—Offset 8160h 978
10.49 GTDRIVER_G2P_EVENTS_0_2_0_GTTMMADR (P_CR_GTDRIV-ER_G2P_EVENTS_0_2_0_GTTMMADR)—Offset 8164h 978
10.50 CORE_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_CORE_FREQUENCY_CAPA-BILITIES_0_2_0_GTTMMADR)—Offset 816Ch 979
10.51 GRAPHICS_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_GRAPHICS_FREQUEN-CY_CAPABILITIES_0_2_0_GTTMMADR)—Offset 8170h 980
10.52 SYSTEM_AGENT_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_SYSTEM_AGENT_-FREQUENCY_CAPABILITIES_0_2_0_GTTMMADR)—Offset 8174h 980
10.53 Memory Frequency Status (P_CR_FAR_MEMORY_FREQUENCY_CAPABILI-

TIES_0_2_0_GTTMMADR)—Offset 8178h 981
10.54 GT_PERF_LIMIT_REASONS (P_CR_GT_PERF_LIMIT_REASONS_0_2_0_GTTMMADR)—Offset 8184h 982
10.55 GTDRIVER_HWP_REQUEST_0_2_0_GTTMMADR (P_CR_GTDRIVER_HWP_REQUEST_0_2_0_GTTMMADR)—Offset 8190h 984
10.56 GT_VIDEO_BUSYNESS_0_2_0_GTTMMADR (P_CR_GT_VIDEO_BUSYNESS_0_2_0_GTTMMADR)—Offset 819Ch 985
10.57 GEN_REGRW1 - General Purpose register (GEN_REGRW1)—Offset 600h 985
10.58 GEN_REGRW2 - General Purpose register (GEN_REGRW2)—Offset 604h 986
10.59 GEN_REGRW3 - General Purpose register (GEN_REGRW3)—Offset 608h 986
10.60 GEN_REGRW4 - General Purpose register (GEN_REGRW4)—Offset 60Ch 986
10.61 Power Management 1 Control (PM1_CNT)—Offset 4h 990
10.62 Power Management 1 Timer (PM1_TMR)—Offset 8h 991
10.63 General Purpose Event 0 Status (GPE0a_STS)—Offset 20h 992
10.64 General Purpose Event 0 Status (GPE0b_STS)—Offset 24h 995
10.65 General Purpose Event 0 Status (GPE0c_STS)—Offset 28h 995
10.66 General Purpose Event 0 Status (GPE0d_STS)—Offset 2Ch 996
10.67 General Purpose Event 0 Enables (GPE0a_EN)—Offset 30h 996
10.68 General Purpose Event 0 Enable (GPE0b_EN)—Offset 34h 999
10.69 General Purpose Event 0 Enable (GPE0c_EN)—Offset 38h 999
10.70 General Purpose Event 0 Enable (GPE0d_EN)—Offset 3Ch 999
10.71 SMI Control and Enable (SMI_EN)—Offset 40h 1000
10.72 SMI Status Register (SMI_STS)—Offset 44h 1002
10.73 Device Trap Status (DEVTRAP_STS)—Offset 4Ch 1006
10.74 General Purpose Event Control (GPE_CTRL)—Offset 50h 1006
10.75 TCO Reload Register (TCO_RLD)—Offset 60h 1007
10.76 TCO Timer Status (TCO_STS)—Offset 64h 1007
10.77 TCO Timer Control (TCO1_CNT)—Offset 68h 1008
10.78 TCO Timer Register (TCO_TMR)—Offset 70h 1008
10.79 Advanced Power Management Status (APM_STS)—Offset 74h 1009
10.80 Advanced Power Management Control Port (APM_CNT)—Offset 78h 1009
10.81 Direct IRQ Enables (DIRECT_IRQ_EN)—Offset 7Ch 1010
10.82 Power and Reset Status (PRSTS)—Offset 1000h 1011
10.83 PM CFG - Power Management Configuration (PMC_CFG)—Offset 1008h 1012
10.84 Power Management Configuration (PMC_CFG2)—Offset 100Ch 1015
10.85 SOC Power Management Status (SOC_PM_STS)—Offset 1010h 1016
10.86 General PM Configuration 1 (GEN_PMCON1)—Offset 1020h 1016
10.87 General PM Configuration 2 (GEN_PMCON2)—Offset 1024h 1019
10.88 General PM Configuration 3 (GEN_PMCON3)—Offset 1028h 1020
10.89 Configured Revision ID (CRID)—Offset 1030h 1023
10.90 Function Disable 0 (FUNC_DIS_0)—Offset 1034h 1023
10.91 Function Disable 1 (FUNC_DIS_1)—Offset 1038h 1025
10.92 Extended Test Mode Register (ETR)—Offset 1048h 1026
10.93 GPIO Group to General Purpose Event Register Configuration (GPIO_GPE_CFG)—Offset 1050h 1027
10.94 UART IRQ Configuration (IRQ_CFG_UART)—Offset 1060h 1028
10.95 IRQ Select 0 (IRQ_SEL_0)—Offset 1064h 1029
10.96 IRQ Select 1 (IRQ_SEL_1)—Offset 1068h 1030
10.97 IRQ Select 2 (IRQ_SEL_2)—Offset 106Ch 1030
10.98 Function ACPI Enumeration 0 (FUNC_ACPI_ENUM_0)—Offset 1070h 1031
10.99 Function ACPI Enumeration 1 (FUNC_ACPI_ENUM_1)—Offset 1074h 1033
10.100 Fixed Deep S0Ix Counter Lower 32 Bits (TELEM_DEEP_S0IX_LO_HOST)—Offset 1078h . 1033
10.101 Fixed Deep S0Ix Counter Upper 32 Bits (TELEM_DEEP_S0IX_HI_HOST)—Offset 107Ch . 1034

10.102 Fixed Shallow S0Ix Counter Lower 32 Bits (TELEM_SHALLOW_S0IX_LO_HOST)—Offset 1080h 1034
10.103 Fixed Shallow S0Ix Counter Upper 32 Bits (TELEM_SHALLOW_S0IX_HI_HOST)—Offset 1084h 1034
10.104 Reserved (TELEM_MISC_FIXED_LO_HOST)—Offset 1088h 1034
10.105 Reserved register for sharing data with software upper 32 bits (TELEM_MISC_-FIXED_HI_HOST)—Offset 108Ch 1035
10.106 BIOS scratchpad (BIOS_SCRATCHPAD)—Offset 1090h 1035
10.107 Display Hot Plug Detect Control (DISPLAY_HPD_CTL)—Offset 1094h 1037
10.108 U2U Bridge Survivability Register 1 (U2UB_SR_1)—Offset 10A0h 1038
10.109 U2U Bridge Survivability Register 2 (U2UB_SR_2)—Offset 10A4h 1040
10.110 OBFF Control and Status (OBFF_CTL_STS)—Offset 10C8h 1041
10.111 Lock Register (LOCK)—Offset 10CCh 1042
10.112 IPC Status (IPC_STS)—Offset 4h 1045
10.113 IPC Source Pointer (IPC_SPTR)—Offset 8h 1046
10.114 IPC Destination Pointer (IPC_DPTR)—Offset Ch 1046
10.115 IPC Write Buffer (IPC_WBUF0)—Offset 80h 1047
10.116 IPC Write Buffer (IPC_WBUF1)—Offset 84h 1047
10.117 IPC Write Buffer (IPC_WBUF2)—Offset 88h 1047
10.118 IPC Write Buffer (IPC_WBUF3)—Offset 8Ch 1048
10.119 IPC Read Buffer (IPC_RBUF0)—Offset 90h 1048
10.120 IPC Read Buffer (IPC_RBUF1)—Offset 94h 1048
10.121 IPC Read Buffer (IPC_RBUF2)—Offset 98h 1049
10.122 IPC Read Buffer (IPC_RBUF3)—Offset 9Ch 1049
10.123 PWM D0i3 Control Register (PWMD0i3C)—Offset 100h 1050
10.124 PWM Control Register (PWMCTRL_1)—Offset 400h 1051
10.125 PWM Control Register (PWMCTRL_2)—Offset 800h 1052
10.126 PWM Control Register (PWMCTRL_3)—Offset C00h 1053

11 Primary to Sideband Bridge Registers 1055

11.1 PCI Identifier (PCIID) — Offset 0h 1055
11.2 PCI Command (PCICMD) — Offset 4h 1055
11.3 PCI Status (PCISTS) — Offset 6h 1056
11.4 Revision ID (PCIRID) — Offset 8h 1056
11.5 Class Code (PCICC) — Offset 9h 1057
11.6 PCI Master Latency Timer (PCIMLT) — Offset Dh 1057
11.7 PCI Header Type (PCIHTYPE) — Offset Eh 1057
11.8 Sideband Register Access BAR (SBREG_BAR) — Offset 10h 1058
11.9 Sideband Register BAR High DWORD (SBREG_BARH) — Offset 14h 1058
11.10 PCI Subsystem Identifiers (PCIHSS) — Offset 2Ch 1058
11.11 VLW Bus:Device:Function (VBDF) — Offset 50h 1059
11.12 ERROR Bus:Device:Function (EBDF) — Offset 52h 1059
11.13 Routing Configuration (RCFG) — Offset 54h 1060
11.14 High Performance Event Timer Configuration (HPTC) — Offset 60h 1060
11.15 IOxAPIC Configuration (IOAC) — Offset 64h 1060
11.16 IOxAPIC Bus:Device:Function (IBDF) — Offset 6Ch 1061
11.17 HPET Bus:Device:Function (HBDF) — Offset 70h 1061
11.18 Sideband Register posted 0 (SBREGPOSTED0) — Offset 80h 1062
11.19 Sideband Register posted 1 (SBREGPOSTED1) — Offset 84h 1062
11.20 Sideband Register posted 2 (SBREGPOSTED2) — Offset 88h 1062
11.21 Sideband Register posted 3 (SBREGPOSTED3) — Offset 8Ch 1063
11.22 Sideband Register posted 4 (SBREGPOSTED4) — Offset 90h 1063
11.23 Sideband Register posted 5 (SBREGPOSTED5) — Offset 94h 1063
11.24 Sideband Register posted 6 (SBREGPOSTED6) — Offset 98h 1064
11.25 Sideband Register posted 7 (SBREGPOSTED7) — Offset 9Ch 1064

11.26 Display Bus:Device:Function (DISPBDF) — Offset A0h 1064
11.27 ICC Register Offsets (ICCOS) — Offset A4h 1065
11.28 Endpoint Mask 0 (EPMASK0) — Offset B0h 1065
11.29 Endpoint Mask 1 (EPMASK1) — Offset B4h 1066
11.30 Endpoint Mask 2 (EPMASK2) — Offset B8h 1066
11.31 Endpoint Mask 3 (EPMASK3) — Offset BCh 1066
11.32 Endpoint Mask 4 (EPMASK4) — Offset C0h 1067
11.33 Endpoint Mask 5 (EPMASK5) — Offset C4h 1067
11.34 Endpoint Mask 6 (EPMASK6) — Offset C8h 1067
11.35 Endpoint Mask 7 (EPMASK7) — Offset CCh 1067
11.36 SBI Address (SBIADDR) — Offset D0h 1068
11.37 SBI Data (SBIDATA) — Offset D4h 1068
11.38 SBI Status (SBISTAT) — Offset D8h 1069
11.39 SBI Routing Identification (SBIRID) — Offset DAh 1069
11.40 SBI Extended Address (SBIEXTADDR) — Offset DCh 1070
11.41 P2SB Control (P2SBC) — Offset E0h 1070
11.42 Power Control Enable (PCE) — Offset E4h 1071
11.43 Power Gate Control Block Timing (PGCBT) — Offset E6h 1071
11.44 Primary Clock Domain Controls (PDOMAIN) — Offset E8h 1072
11.45 Sideband Clock Domain Controls (SDOMAIN) — Offset EAh 1072
11.46 Unsupported Request Error Status (URES) — Offset F0h 1073
11.47 Unsupported Request Error Control (UREC) — Offset F4h 1073
11.48 Manufacturers ID (MANID) — Offset F8h 1074
11.49 SAI Policy Control (SAIPOLCTRL0) — Offset 240h 1074
11.50 SAI Policy Control (SAIPOLCTRL1) — Offset 244h 1075
11.51 SAI Policy Access Control (SAIPOLAC0) — Offset 248h 1075
11.52 SAI Policy Access Control (SAIPOLAC1) — Offset 24Ch 1075

12 PWM Registers 1077

12.1 DEVICEVENDORID - Device ID and Vendor ID Register (DEVVENDID) — Offset 0h . 1077
12.2 STATUSCOMMAND — Offset 4h 1077
12.3 REVCLASSCODE — Offset 8h 1078
12.4 CLLATHEADERBIST — Offset Ch 1078
12.5 BAR — Offset 10h 1079
12.6 BAR -Base Address Register High (BAR_HIGH) — Offset 14h 1079
12.7 BAR1 — Offset 18h 1080
12.8 BAR1 -Base Address Register1 High (BAR1_HIGH) — Offset 1Ch 1080
12.9 BAR2 — Offset 20h 1081
12.10 SUBSYSTEMID — Offset 2Ch 1081
12.11 EXPANSION ROM base address (EXPANSION_ROM_BASEADDR) — Offset 30h 1081
12.12 CAPABILITYPTR — Offset 34h 1082
12.13 INTERRUPTREG — Offset 3Ch 1082
12.14 POWERCAPID — Offset 80h 1082
12.15 PMECTRLSTATUS — Offset 84h 1083
12.16 PCI DEVICE IDLE CAPABILITY RECORD (PCIDEVIDLE_CAP_RECORD) — Offset 90h. 1083
12.17 DEVID VENDOR SPECIFIC REG (DEVID_VEND_SPECIFIC_REG) — Offset 94h 1084
12.18 D0I3_CONTROL_SW_LTR_MMIO_REG — Offset 98h 1084
12.19 DEVICE_IDLE_POINTER_REG — Offset 9Ch 1085
12.20 D0I3_MAX_POW_LAT_PG_CONFIG — Offset A0h 1085
12.21 GEN_REGRW1 - General Purpose Read Write Register1 (GEN_REGRW1) — Offset B0h ... 1086
12.22 GEN_REGRW2 — Offset B4h 1086
12.23 GEN_REGRW3 — Offset B8h 1086
12.24 GEN_REGRW4 — Offset BCh 1087
12.25 GEN_INPUT_REG — Offset C0h 1087

12.26 Manufacturers ID (MANID) — Offset F8h 1087

13 SD and MMC Registers 1089

13.1 DEVVENDID — Offset 0h 1089
13.2 STATUSCOMMAND — Offset 4h 1089
13.3 REVCLASSCODE — Offset 8h 1090
13.4 CLLATHEADERBIST — Offset Ch 1090
13.5 BAR — Offset 10h....1091
13.6 BAR_HIGH — Offset 14h 1091
13.7 BAR1 — Offset 18h....1092
13.8 BAR1_HIGH — Offset 1Ch 1092
13.9 SUBSYSTEMID — Offset 2Ch....1092
13.10 EXPANSION_ROM_BASEADDR — Offset 30h 1093
13.11 CAPABILITYPTR — Offset 34h 1093
13.12 INTERRUPTREG — Offset 3Ch 1093
13.13 POWERCAPID — Offset 80h 1094
13.14 PMECTRLSTATUS — Offset 84h 1094
13.15 PCIDEVIDLE_CAP_RECORD — Offset 90h 1095
13.16 DEVID_VEND_SPECIFIC_REG — Offset 94h 1095
13.17 D0I3_CONTROL_SW_LTR_MMIO_REG — Offset 98h....1096
13.18 DEVICE_IDLE_POINTER_REG — Offset 9Ch....1096
13.19 D0I3_MAX_POW_LAT_PG_CONFIG — Offset A0h 1096
13.20 GEN_REGRW1 — Offset B0h 1097
13.21 GEN_REGRW2 — Offset B4h 1097
13.22 GEN_REGRW3 — Offset B8h 1098
13.23 GEN_REGRW4 — Offset BCh 1098
13.24 GEN_INPUT_REG — Offset C0h 1098
13.25 MANID — Offset F8h....1099
13.26 SW_LTR_Value (SW_LTR_val)—Offset 804h....1099
13.27 Auto LTR Value (Auto_LTR_val)—Offset 808h....1100
13.28 Capabilities Bypass Control (Cap_byps)—Offset 810h 1100
13.29 emmc Capabilities Bypass Register I (Cap_byps_reg1)—Offset 814h 1101
13.30 emmc Capabilities Bypass Register 2 (Cap_byps_reg2)—Offset 818h....1102
13.31 D0i3 (reg_D0i3)—Offset 81Ch....1103
13.32 Tx CMD Delay Control (Tx_CMD_dly)—Offset 820h 1103
13.33 Tx DATA Delay Control 1 (Tx_DATA_dly_1)—Offset 824h....1104
13.34 Tx DATA Delay Control 2 (Tx_DATA_dly_2)—Offset 828h....1104
13.35 Rx CMD Data Delay Control 1 (Rx_CMD_Data_dly_1)—Offset 82Ch....1105
13.36 Rx Strobe Delay Control (Rx_Strobe_Ctrl_Path)—Offset 830h....1106
13.37 Rx CMD Data Delay Control 2 (Rx_CMD_Data_dly_2)—Offset 834h....1106
13.38 Master DLL Software Control (Master_Dll)—Offset 838h....1107
13.39 Auto_tuning_val (Auto_tuning)—Offset 840h 1108
13.40 Root Space sel (emmc_Root_Space)—Offset 900h 1108
13.41 SW_LTR_Value (SW_LTR_val)—Offset 804h....1109
13.42 Auto LTR Value (Auto_LTR_val)—Offset 808h....1109
13.43 Capabilities Bypass Control (Cap_byps)—Offset 810h 1110
13.44 emmc Capabilities Bypass Register I (Cap_byps_reg1)—Offset 814h 1110
13.45 emmc Capabilities Bypass Register 2 (Cap_byps_reg2)—Offset 818h....1112
13.46 D0i3 (reg_D0i3)—Offset 81Ch....1112
13.47 Tx CMD Delay Control (Tx_CMD_dly)—Offset 820h 1113
13.48 Tx DATA Delay Control 1 (Tx_DATA_dly_1)—Offset 824h....1114
13.49 Tx DATA Delay Control 2 (Tx_DATA_dly_2)—Offset 828h....1114
13.50 Rx CMD Data Delay Control 1 (Rx_CMD_Data_dly_1)—Offset 82Ch....1115
13.51 Rx CMD Data Delay Control 2 (Rx_CMD_Data_dly_2)—Offset 834h....1116
13.52 Auto_tuning_val (Auto_tuning)—Offset 840h 1116

13.53 SDMA System Address Register/Argument2 Register (sdmasysaddr)—Offset 0h 1117
13.54 BlockSize Register (blocksize)—Offset 4h 1117
13.55 BlockCount Register (blockcount)—Offset 6h 1117
13.56 Argument1 Register (argument1)—Offset 8h 1118
13.57 TransferMode Register (transfermode)—Offset Ch 1118
13.58 Command Register (command)—Offset Eh 1119
13.59 Response Register (response01)—Offset 10h 1119
13.60 Response Register (response2)—Offset 14h 1120
13.61 Response Register (response3)—Offset 16h 1120
13.62 Response Register (response4)—Offset 18h 1120
13.63 Response Register (response5)—Offset 1Ah 1121
13.64 Response Register (response6)—Offset 1Ch 1121
13.65 Response Register (response7)—Offset 1Eh 1121
13.66 Buffer DataPort Register (dataport)—Offset 20h 1122
13.67 PresentState Register (presentstate)—Offset 24h 1122
13.68 HostControl1 Register (hostcontrol1)—Offset 28h 1124
13.69 PowerControl Register (powercontrol)—Offset 29h 1124
13.70 BlockGapControl Register (blockgapcontrol)—Offset 2Ah 1125
13.71 Wakeup Control Register (wakeupcontrol)—Offset 2Bh 1125
13.72 Clock Control Register (clockcontrol)—Offset 2Ch 1126
13.73 Timeout Control Register (timeoutcontrol)—Offset 2Eh 1126
13.74 Software Reset Register (softwarereset)—Offset 2Fh 1127
13.75 Normal Interrupt Status Register (normalintrsts)—Offset 30h 1127
13.76 ErrorInterruptStatus_Register (errorintrsts)—Offset 32h 1129
13.77 Normal Interrupt Status Enable Register (normalintrstsena)—Offset 34h 1130
13.78 Error Interrupt Status Enable Register (errorintrstsena)—Offset 36h 1131
13.79 Normal Interrupt Signal Enable Register (normalintrsigena)—Offset 38h 1132
13.80 Error Interrupt Signal Enable Register (errorintrsigena)—Offset 3Ah 1133
13.81 Auto CMD12 Error Status Register (autocmderrsts)—Offset 3Ch 1134
13.82 Host Control2 Register (hostcontrol2)—Offset 3Eh 1135
13.83 Capabilities Register (capabilities)—Offset 40h 1136
13.84 Maximum Current Capabilities Register (maxcurrentcap)—Offset 48h 1138
13.85 Force Event REGISTER for AUTO CMD Error Status (ForceEventforAUTOCMDErrorStatus)—Offset 50h 1139
13.86 Force Event Register for Error Interrupt Status (forceeventforerrintsts)—Offset 52h 1139
13.87 ADMA Error Status Register (admaerrsts)—Offset 54h 1140
13.88 ADMA System Address Register0&1 (admasysaddr01)—Offset 58h 1141
13.89 ADMA System Address Register1 (admasysaddr2)—Offset 5Ch 1141
13.90 ADMA System Address Register1 (admasysaddr3)—Offset 5Eh 1141
13.91 Preset Value Register for Initialization (presetvalue0)—Offset 60h 1142
13.92 Preset Value Register for Default Speed (presetvalue1)—Offset 62h 1142
13.93 Preset Value Register for High Speed (presetvalue2)—Offset 64h 1143
13.94 Preset Value Register for SDR12 (presetvalue3)—Offset 66h 1143
13.95 Preset Value Register for SDR25 (presetvalue4)—Offset 68h 1144
13.96 Preset Value Register for SDR50 (presetvalue5)—Offset 6Ah 1144
13.97 Preset Value Register for SDR104 (presetvalue6)—Offset 6Ch 1145
13.98 Preset Value Register for DDR50 (presetvalue7)—Offset 6Eh 1145
13.99 Boot Timeout Control Register (boottimeoutcnt)—Offset 70h 1146
13.100Preset Value Register for DDR50 (presetvalue8)—Offset 74h 1146
13.101Vendor Register (vendreg)—Offset 78h 1147
13.102Slot Interrupt Status Register (slotintrsts)—Offset FCh 1147
13.103Host Controller Version Register (hostcontrollerver)—Offset FEh 1148
13.104CQVER (CQVER)—Offset 200h 1148
13.105CQCAP (CQCAP)—Offset 204h 1148
13.106CQCFG (CQCFG)—Offset 208h 1149

13.107CQCTL (CQCTL)—Offset 20Ch 1150
13.108Command Queueing Interrupt Status (CQIS)—Offset 210h 1150
13.109Command Queueing Interrupt Status Enabled (CQISTE)—Offset 214h 1151
13.110Command Queueing Interrupt Status Enable (CQISGE)—Offset 218h 1152
13.111Interrupt Coalescing (CQIC)—Offset 21Ch 1152
13.112Command Queueing Task Descriptor List Base Address (CQTDLBA)—Offset 220h ... 1153
13.113Command Queueing Task Descriptor List Base Address Upper 32 Bits (CQTDLBAU)—Offset 224h 1154
13.114Command Queueing Taks Doorbell (CQTDRB)—Offset 228h 1154
13.115Command Queueing Taks Doorbell Notification (CQTCN)—Offset 22Ch 1154
13.116Command Queueing Device Queue Status (CQDQS)—Offset 230h 1155
13.117Command Queueing Device Pending Tasks (CQDPT)—Offset 234h 1155
13.118Command Queueing Task Clear (CQTCLR)—Offset 238h 1155
13.119Send Status Timer Configuration 1 (CQSSC1)—Offset 240h 1156
13.120Send Status Configuration 2 (CQSSC2)—Offset 244h 1156
13.121Command Response for Direct-Command Task (CQCRDCT)—Offset 248h 1157
13.122Response Mode Error Mask (CQRMEM)—Offset 250h 1157
13.123CQBASE+54h:CQTERRI - Task Error Information (CQTERRI)—Offset 254h 1158
13.124Command Response Index (CQCRI)—Offset 258h 1158
13.125Command Response Argument (CQCRA)—Offset 25Ch 1159

14 LPIO Registers 1160

14.1 DEVICEVENDORID - Device ID and Vendor ID Register (DEVVENDID) — Offset 0h .. 1160
14.2 STATUSCOMMAND — Offset 4h 1161
14.3 REVCLASSCODE — Offset 8h 1161
14.4 CLLATHEADERBIST — Offset Ch 1162
14.5 BAR — Offset 10h 1162
14.6 BAR -Base Address Register High (BAR_HIGH) — Offset 14h 1163
14.7 BAR1 — Offset 18h 1163
14.8 BAR1 -Base Address Register1 High (BAR1_HIGH) — Offset 1Ch 1164
14.9 SUBSYSTEMID — Offset 2Ch 1164
14.10 EXPANSION ROM base address (EXPANSION_ROM_BASEADDR) — Offset 30h 1164
14.11 CAPABILITYPTR — Offset 34h 1165
14.12 INTERRUPTREG — Offset 3Ch 1165
14.13 POWERCAPID — Offset 80h 1166
14.14 PMECTRLSTATUS — Offset 84h 1166
14.15 PCI DEVICE IDLE CAPABILITY RECORD (PCIDEVIDLE_CAP_RECORD) — Offset 90h . 1167
14.16 DEVID VENDOR SPECIFIC REG (DEVID_VEND_SPECIFIC_REG) — Offset 94h 1167
14.17 D0I3_CONTROL_SW_LTR_MMIO_REG — Offset 98h 1168
14.18 DEVICE_IDLE_POINTER_REG — Offset 9Ch 1168
14.19 D0I3_MAX_POW_LAT_PG_CONFIG — Offset A0h 1168
14.20 GEN_REGRW1 - General Purpose Read Write Register1 (GEN_REGRW1) — Offset B0h 1169
14.21 GEN_REGRW2 — Offset B4h 1169
14.22 GEN_REGRW3 — Offset B8h 1170
14.23 GEN_REGRW4 — Offset BCh 1170
14.24 GEN_INPUT_REG — Offset C0h 1170
14.25 MANID — Offset F8h 1171

15 LPIO BAR Registers 1172

15.1 IC_CON — Offset 0h 1172
15.2 IC_TAR — Offset 4h 1173
15.3 IC_SAR — Offset 8h 1173
15.4 IC_HS_MADDR — Offset Ch 1174
15.5 IC_DATA_CMD — Offset 10h 1174
15.6 IC_SS_SCL_HCNT — Offset 14h 1174

15.7 IC_SS_SCL_LCNT — Offset 18h 1175
15.8 IC_FS_SCL_HCNT — Offset 1Ch 1175
15.9 IC_FS_SCL_LCNT — Offset 20h 1175
15.10 IC_HS_SCL_HCNT — Offset 24h 1176
15.11 IC_HS_SCL_LCNT — Offset 28h 1176
15.12 IC_INTR_STAT — Offset 2Ch 1176
15.13 IC_INTR_MASK — Offset 30h 1177
15.14 IC_RAW_INTR_STAT — Offset 34h 1178
15.15 IC_RX_TL — Offset 38h 1179
15.16 IC_TX_TL — Offset 3Ch 1179
15.17 IC_CLR_INTR — Offset 40h 1180
15.18 IC_CLR_RX_UNDER — Offset 44h 1180
15.19 IC_CLR_RX_OVER — Offset 48h 1180
15.20 IC_CLR_TX_OVER — Offset 4Ch 1181
15.21 IC_CLR_RD_REQ — Offset 50h 1181
15.22 IC_CLR_TX_ABRT — Offset 54h 1181
15.23 IC_CLR_RX_DONE — Offset 58h 1182
15.24 IC_CLR_ACTIVITY — Offset 5Ch 1182
15.25 IC_CLR_STOP_DET — Offset 60h 1182
15.26 IC_CLR_START_DET — Offset 64h 1183
15.27 IC_CLR_GEN_CALL — Offset 68h 1183
15.28 IC_ENABLE — Offset 6Ch 1183
15.29 IC_STATUS — Offset 70h 1184
15.30 IC_TXFLR — Offset 74h 1184
15.31 IC_RXFLR — Offset 78h 1185
15.32 IC_SDA_HOLD — Offset 7Ch 1185
15.33 IC_TX_ABRT_SOURCE — Offset 80h 1185
15.34 IC_SLV_DATA_NACK_ONLY — Offset 84h 1186
15.35 IC_DMA_CR — Offset 88h 1187
15.36 IC_DMA_TDLR — Offset 8Ch 1187
15.37 IC_DMA_RDLR — Offset 90h 1187
15.38 IC_SDA_SETUP — Offset 94h 1188
15.39 IC_ACK_GENERAL_CALL — Offset 98h 1188
15.40 IC_ENABLE_STATUS — Offset 9Ch 1188
15.41 IC_FS_SPKLEN — Offset A0h 1189
15.42 IC_HS_SPKLEN — Offset A4h 1189
15.43 IC_CLR_RESTART_DET (IC_CLR_RESTRART_DET) — Offset A8h 1190
15.44 IC_COMP_PARAM_1 — Offset F4h 1190
15.45 IC_COMP_VERSION — Offset F8h 1190
15.46 IC_COMP_TYPE — Offset FCh 1191
15.47 reg_Reserved0 (RESERVED0) — Offset 200h 1191
15.48 reg_RESETS (RESETS) — Offset 204h 1191
15.49 reg_GENERAL (GENERAL) — Offset 208h 1192
15.50 reg_ACTIVELTR (ACTIVELTR_VALUE) — Offset 210h 1193
15.51 reg_IDLELTR (IDLELTR_VALUE) — Offset 214h 1193
15.52 reg_TX_ACK_COUNT (TX_ACK_COUNT) — Offset 218h 1194
15.53 rx_ack_count (RX_BYTE_COUNT) — Offset 21Ch 1194
15.54 TX_COMPLETE_INTR_STAT — Offset 220h 1195
15.55 TX_COMPLETE_INTR_CLR — Offset 224h 1195
15.56 reg_SW_SCRATCH_0 (SW_SCRATCH_0) — Offset 228h 1195
15.57 reg_SW_SCRATCH_1 (SW_SCRATCH_1) — Offset 22Ch 1196
15.58 reg_SW_SCRATCH_2 (SW_SCRATCH_2) — Offset 230h 1196
15.59 reg_SW_SCRATCH_3 (SW_SCRATCH_3) — Offset 234h 1196
15.60 reg_CLOCK_GATE (CLOCK_GATE) — Offset 238h 1196
15.61 reg_Reserved1 (RESERVED1) — Offset 23Ch 1197

15.62 reg_REMAP_ADDR_LO (REMAP_ADDR_LO) — Offset 240h 1197
15.63 reg_REMAP_ADDR_HI (REMAP_ADDR_HI) — Offset 244h 1197
15.64 reg_Reserved2 (RESERVED2) — Offset 248h....1198
15.65 reg_DEVIDLE_CONTROL (DEVIDLE_CONTROL) — Offset 24Ch....1198
15.66 reg_CAPABLITIES (CAPABLITIES) — Offset 2FCh 1198
15.67 SAR_LO0 — Offset 800h....1199
15.68 SAR_HI0 — Offset 804h 1200
15.69 DAR_LO0 — Offset 808h 1200
15.70 DAR_HI0 — Offset 80Ch....1201
15.71 LLP_LO0 — Offset 810h....1201
15.72 LLP_HI0 — Offset 814h 1202
15.73 CTL_LO0 — Offset 818h 1202
15.74 CTL_HI0 — Offset 81Ch 1204
15.75 SSTAT0 — Offset 820h....1204
15.76 DSTAT0 — Offset 828h 1205
15.77 SSTATAR_LO0 — Offset 830h 1205
15.78 SSTATAR_HI0 — Offset 834h 1206
15.79 DSTATAR_LO0 — Offset 838h 1206
15.80 DSTATAR_HI0 — Offset 83Ch....1206
15.81 CFG_LO0 — Offset 840h....1207
15.82 CFG_HI0 — Offset 844h 1208
15.83 SGR0 — Offset 848h....1209
15.84 DSR0 — Offset 850h....1209
15.85 SAR_LO1 — Offset 858h....1210
15.86 SAR_HI1 — Offset 85Ch....1210
15.87 DAR_LO1 — Offset 860h 1211
15.88 DAR_HI1 — Offset 864h....1211
15.89 LLP_LO1 — Offset 868h....1212
15.90 LLP_HI1 — Offset 86Ch....1213
15.91 CTL_LO1 — Offset 870h 1213
15.92 CTL_HI1 — Offset 874h....1214
15.93 SSTAT1 — Offset 878h....1215
15.94 DSTAT1 — Offset 880h 1216
15.95 SSTATAR_LO1 — Offset 888h 1216
15.96 SSTATAR_HI1 — Offset 88Ch....1216
15.97 DSTATAR_LO1 — Offset 890h 1217
15.98 DSTATAR_HI1 — Offset 894h....1217
15.99 CFG_LO1 — Offset 898h....1217
15.100CFG_HI1 — Offset 89Ch 1219
15.101SGR1 — Offset 8A0h 1219
15.102DSR1 — Offset 8A8h 1220
15.103RAWTFR — Offset AC0h....1220
15.104RAWBLOCK — Offset AC8h....1221
15.105RAWSRCTRAN — Offset AD0h....1221
15.106RAWDSTTRAN — Offset AD8h....1222
15.107RAWERR — Offset AE0h....1222
15.108STATUSTFR — Offset AE8h....1223
15.109STATUSBLOCK — Offset AF0h 1223
15.110STATUSSRCTRAN — Offset AF8h....1223
15.111STATUSDSTTRAN — Offset B00h 1224
15.112STATUSERR — Offset B08h 1224
15.113MASKTFR — Offset B10h....1225
15.114MASKBLOCK — Offset B18h 1225
15.115MASKSRCTRAN — Offset B20h....1226
15.116MASKDSTTRAN — Offset B28h....1227

15.117MASKERR — Offset B30h 1228
15.118CLEARTFR — Offset B38h 1229
15.119CLEARBLOCK — Offset B40h 1229
15.120CLEARSRCTRAN — Offset B48h 1229
15.121CLEARDSTTRAN — Offset B50h 1230
15.122CLEARERR — Offset B58h 1230
15.123STATUSINT — Offset B60h 1231
15.124DMACFGREG — Offset B98h 1231
15.125CHENREG — Offset BA0h 1232
15.126RESERVED0_CPL — Offset BB8h 1232
15.127RESERVED0_CPH — Offset BBCh 1232
15.128RESERVED1_CPL — Offset BC0h 1233
15.129RESERVED1_CPH — Offset BC4h 1233
15.130RESERVED0_FPL — Offset C00h 1233
15.131RESERVED0_FPH — Offset C04h 1234
15.132RESERVED1_FPL — Offset C08h 1234
15.133RESERVED1_FPH — Offset C0Ch 1234
15.134SAI_ERR — Offset C10h 1235
15.135GLOBAL_CFG — Offset C18h 1235

16 USB-Host (xHCI) Registers 1236

16.1 Vendor ID (VID) — Offset 0h 1236
16.2 Device ID (DID) — Offset 2h 1236
16.3 Command (CMD) — Offset 4h 1236
16.4 Device Status (STS) — Offset 6h 1237
16.5 Revision ID (RID) — Offset 8h 1238
16.6 Programming Interface (PI) — Offset 9h 1238
16.7 Sub Class Code (SCC) — Offset Ah 1239
16.8 Base Class Code (BCC) — Offset Bh 1239
16.9 Master Latency Timer (MLT) — Offset Dh 1239
16.10 Header Type (HT) — Offset Eh 1239
16.11 Memory Base Address (MBAR) — Offset 10h 1240
16.12 USB Subsystem Vendor ID (SSVID) — Offset 2Ch 1240
16.13 USB Subsystem ID (SSID) — Offset 2Eh 1240
16.14 Capabilities Pointer (CAP_PTR) — Offset 34h 1241
16.15 Interrupt Line (ILINE) — Offset 3Ch 1241
16.16 Interrupt Pin (IPIN) — Offset 3Dh 1241
16.17 XHC System Bus Configuration 1 (XHCC1) — Offset 40h 1242
16.18 XHC System Bus Configuration 2 (XHCC2) — Offset 44h 1243
16.19 Clock Gating (XHCLKGTEN) — Offset 50h 1244
16.20 Audio Time Synchronization (AUDSYNC) — Offset 58h 1246
16.21 Serial Bus Release Number (SBRN) — Offset 60h 1246
16.22 Frame Length Adjustment (FLADJ) — Offset 61h 1246
16.23 Best Effort Service Latency (BESL) — Offset 62h 1247
16.24 PCI Power Management Capability ID (PM_CID) — Offset 70h 1247
16.25 Next Item Pointer #1 (PM_NEXT) — Offset 71h 1247
16.26 Power Management Capabilities (PM_CAP) — Offset 72h 1248
16.27 Power Management Control/Status (PM_CS) — Offset 74h 1249
16.28 Message Signaled Interrupt CID (MSI_CID) — Offset 80h 1249
16.29 Next item pointer (MSI_NEXT) — Offset 81h 1250
16.30 Message Signaled Interrupt Message Control (MSI_MCTL) — Offset 82h 1250
16.31 Message Signaled Interrupt Message Address (MSI_MAD) — Offset 84h 1250
16.32 Message Signaled Interrupt Upper Address (MSI_MUAD) — Offset 88h 1251
16.33 Message Signaled Interrupt Message Data (MSI_MD) — Offset 8Ch 1251
16.34 Device Idle Capability (DEVIDLE) — Offset 90h 1251

16.35 Vendor Specific Header (VSHDR) — Offset 94h 1252
16.36 SW LTR POINTER (SWLTRPTR) — Offset 98h 1252
16.37 Device Idle Pointer Register (DEVIDLEPTR) — Offset 9Ch 1253
16.38 Device Idle Power ON Latency (DEVIDLEPOL) — Offset A0h 1253
16.39 High Speed Configuration 2 (HSCFG2) — Offset A4h 1254
16.40 XHCI USB2 Overcurrent Pin Mapping 1 (U2OCM1) — Offset B0h 1254
16.41 XHCI USB2 Overcurrent Pin Mapping 2 (U2OCM2) — Offset B4h 1255
16.42 XHCI USB3 Overcurrent Pin Mapping 1 (U3OCM1) — Offset D0h 1255
16.43 XHCI USB3 Overcurrent Pin Mapping 2 (U3OCM2) — Offset D4h 1255
16.44 XHCC3 — Offset FCh 1256
16.45 Capability Registers Length (CAPLENGTH)—Offset 0h 1256
16.46 Host Controller Interface Version Number (HCIVERSION)—Offset 2h 1256
16.47 Structural Parameters 1 (HCSPARAMS1)—Offset 4h 1257
16.48 Structural Parameters 2 (HCSPARAMS2)—Offset 8h 1257
16.49 Structural Parameters 3 (HCSPARAMS3)—Offset Ch 1258
16.50 Capability Parameters (HCCPARAMS)—Offset 10h 1258
16.51 Doorbell Offset (DBOFF)—Offset 14h 1259
16.52 Runtime Register Space Offset (RTSOFF)—Offset 18h 1259
16.53 USB Command (USBCMD)—Offset 80h 1260
16.54 USB Status (USBSTS)—Offset 84h 1260
16.55 Page Size (PAGESIZE)—Offset 88h 1261
16.56 Device Notification Control (DNCTRL)—Offset 94h 1261
16.57 Command Ring Low (CRCR_LO)—Offset 98h 1262
16.58 Command Ring High (CRCR_HI)—Offset 9Ch 1262
16.59 Device Context Base Address Array Pointer Low (DCBAAP_LO)—Offset B0h 1262
16.60 Device Context Base Address Array Pointer High (DCBAAP_HI)—Offset B4h 1263
16.61 Configure (CONFIG)—Offset B8h 1263
16.62 Port Status and Control USB2 (PORTSC1)—Offset 480h 1263
16.63 Port Power Management Status and Control USB2 (PORTPMSC1)—Offset 484h 1265
16.64 Port X Hardware LPM Control Register (PORTHLPMC1)—Offset 48Ch 1265
16.65 Port Status and Control USB2 (PORTSC2)—Offset 490h 1266
16.66 Port Power Management Status and Control USB2 (PORTPMSC2)—Offset 494h 1267
16.67 Port X Hardware LPM Control Register (PORTHLPMC2)—Offset 49Ch 1268
16.68 Port Status and Control USB2 (PORTSC3)—Offset 4A0h 1269
16.69 Port Power Management Status and Control USB2 (PORTPMSC3)—Offset 4A4h 1270
16.70 Port X Hardware LPM Control Register (PORTHLPMC3)—Offset 4ACh 1270
16.71 Port Status and Control USB2 (PORTSC4)—Offset 4B0h 1271
16.72 Port Power Management Status and Control USB2 (PORTPMSC4)—Offset 4B4h 1272
16.73 Port X Hardware LPM Control Register (PORTHLPMC4)—Offset 4BCh 1273
16.74 Port Status and Control USB2 (PORTSC5)—Offset 4C0h 1274
16.75 Port Power Management Status and Control USB2 (PORTPMSC5)—Offset 4C4h 1275
16.76 Port X Hardware LPM Control Register (PORTHLPMC5)—Offset 4CCh 1276
16.77 Port Status and Control USB2 (PORTSC6)—Offset 4D0h 1276
16.78 Port Power Management Status and Control USB2 (PORTPMSC6)—Offset 4D4h 1278
16.79 Port X Hardware LPM Control Register (PORTHLPMC6)—Offset 4DCh 1278
16.80 Port Status and Control USB2 (PORTSC7)—Offset 4E0h 1279
16.81 Port Power Management Status and Control USB2 (PORTPMSC7)—Offset 4E4h 1280
16.82 Port X Hardware LPM Control Register (PORTHLPMC7)—Offset 4ECh 1281
16.83 Port Status and Control USB2 (PORTSC8)—Offset 4F0h 1281
16.84 Port Power Management Status and Control USB2 (PORTPMSC8)—Offset 4F4h 1283
16.85 Port X Hardware LPM Control Register (PORTHLPMC8)—Offset 4FCh 1283
16.86 Port Status and Control USB2 (PORTSC9)—Offset 500h 1284
16.87 Port Power Management Status and Control USB2 (PORTPMSC9)—Offset 504h 1285
16.88 Port X Hardware LPM Control Register (PORTHLPMC9)—Offset 50Ch 1286
16.89 Port Status and Control USB3 (PORTSC10)—Offset 510h 1287

16.90 Port Power Management Status and Control USB3 (PORTPMSC10)—Offset 514h..... 1288
16.91 USB3 Port Link Info (PORTLI10)—Offset 518h..... 1288
16.92 Port Status and Control USB3 (PORTSC11)—Offset 520h..... 1289
16.93 Port Power Management Status and Control USB3 (PORTPMSC11)—Offset 524h..... 1290
16.94 USB3 Port Link Info (PORTLI11)—Offset 528h..... 1290
16.95 Port Status and Control USB3 (PORTSC12)—Offset 530h..... 1291
16.96 Port Power Management Status and Control USB3 (PORTPMSC12)—Offset 534h..... 1292
16.97 USB3 Port Link Info (PORTLI12)—Offset 538h..... 1292
16.98 Port Status and Control USB3 (PORTSC13)—Offset 540h..... 1293
16.99 Port Power Management Status and Control USB3 (PORTPMSC13)—Offset 544h..... 1294
16.100USB3 Port Link Info (PORTLI13)—Offset 548h 1294
16.101Port Status and Control USB3 (PORTSC14)—Offset 550h 1295
16.102Port Power Management Status and Control USB3 (PORTPMSC14)—Offset 554h 1296
16.103USB3 Port Link Info (PORTLI14)—Offset 558h 1296
16.104Port Status and Control USB3 (PORTSC15)—Offset 560h 1297
16.105Port Power Management Status and Control USB3 (PORTPMSC15)—Offset 564h 1298
16.106USB3 Port Link Info (PORTLI15)—Offset 568h 1298
16.107Port Status and Control USB3 (PORTSC16)—Offset 570h 1299
16.108Port Power Management Status and Control USB3 (PORTPMSC16)—Offset 574h 1300
16.109USB3 Port Link Info (PORTLI16)—Offset 578h 1300
16.110Microframe Index (RTMFINDEX)—Offset 2000h 1301
16.111Interrupter 1 Management (IMAN0)—Offset 2020h 1301
16.112Interrupter 1 Moderation (IMOD0)—Offset 2024h 1301
16.113Event Ring Segment Table Size 1 (ERSTSZ0)—Offset 2028h 1302
16.114Event Ring Segment Table Base Address Low 1 (ERSTBA_LO0)—Offset 2030h..... 1302
16.115Event Ring Segment Table Base Address High 1 (ERSTBA_HI0)—Offset 2034h 1302
16.116Event Ring Dequeue Pointer Low 1 (ERDP_LO0)—Offset 2038h..... 1303
16.117Event Ring Dequeue Pointer High 1 (ERDP_HI0)—Offset 203Ch 1303
16.118Interrupter 2 Management (IMAN1)—Offset 2040h 1303
16.119Interrupter 2 Moderation (IMOD1)—Offset 2044h 1304
16.120Event Ring Segment Table Size 2 (ERSTSZ1)—Offset 2048h 1304
16.121Event Ring Segment Table Base Address Low 2 (ERSTBA_LO1)—Offset 2050h..... 1305
16.122Event Ring Segment Table Base Address High 2 (ERSTBA_HI1)—Offset 2054h 1305
16.123Event Ring Dequeue Pointer Low 2 (ERDP_LO1)—Offset 2058h..... 1305
16.124Event Ring Dequeue Pointer High 2 (ERDP_HI1)—Offset 205Ch 1306
16.125Interrupter 3 Management (IMAN2)—Offset 2060h 1306
16.126Interrupter 3 Moderation (IMOD2)—Offset 2064h 1306
16.127Event Ring Segment Table Size 3 (ERSTSZ2)—Offset 2068h 1307
16.128Event Ring Segment Table Base Address Low 3 (ERSTBA_LO2)—Offset 2070h..... 1307
16.129Event Ring Segment Table Base Address High 3 (ERSTBA_HI2)—Offset 2074h 1307
16.130Event Ring Dequeue Pointer Low 3 (ERDP_LO2)—Offset 2078h..... 1308
16.131Event Ring Dequeue Pointer High 3 (ERDP_HI2)—Offset 207Ch 1308
16.132Interrupter 4 Management (IMAN3)—Offset 2080h 1308
16.133Interrupter 4 Moderation (IMOD3)—Offset 2084h 1309
16.134Event Ring Segment Table Size 4 (ERSTSZ3)—Offset 2088h 1309
16.135Event Ring Segment Table Base Address Low 4 (ERSTBA_LO3)—Offset 2090h..... 1310
16.136Event Ring Segment Table Base Address High 4 (ERSTBA_HI3)—Offset 2094h 1310
16.137Event Ring Dequeue Pointer Low 4 (ERDP_LO3)—Offset 2098h..... 1310
16.138Event Ring Dequeue Pointer High 4 (ERDP_HI3)—Offset 209Ch 1311
16.139Interrupter 5 Management (IMAN4)—Offset 20A0h 1311
16.140Interrupter 5 Moderation (IMOD4)—Offset 20A4h 1311
16.141Event Ring Segment Table Size 5 (ERSTSZ4)—Offset 20A8h 1312
16.142Event Ring Segment Table Base Address Low 5 (ERSTBA_LO4)—Offset 20B0h 1312
16.143Event Ring Segment Table Base Address High 5 (ERSTBA_HI4)—Offset 20B4h 1312
16.144Event Ring Dequeue Pointer Low 5 (ERDP_LO4)—Offset 20B8h 1313

16.145Event Ring Dequeue Pointer High 5 (ERDP_HI4)—Offset 20BCh 1313
16.146Interrupter 6 Management (IMAN5)—Offset 20C0h 1313
16.147Interrupter 6 Moderation (IMOD5)—Offset 20C4h.....1314
16.148Event Ring Segment Table Size 6 (ERSTSZ5)—Offset 20C8h.....1314
16.149Event Ring Segment Table Base Address Low 6 (ERSTBA_LO5)—Offset 20D0h.....1315
16.150Event Ring Segment Table Base Address High 6 (ERSTBA_HI5)—Offset 20D4h 1315
16.151Event Ring Dequeue Pointer Low 6 (ERDP_LO5)—Offset 20D8h.....1315
16.152Event Ring Dequeue Pointer High 6 (ERDP_HI5)—Offset 20DCh 1316
16.153Interrupter 7 Management (IMAN6)—Offset 20E0h.....1316
16.154Interrupter 7 Moderation (IMOD6)—Offset 20E4h.....1316
16.155Event Ring Segment Table Size 7 (ERSTSZ6)—Offset 20E8h.....1317
16.156Event Ring Segment Table Base Address Low 7 (ERSTBA_LO6)—Offset 20F0h 1317
16.157Event Ring Segment Table Base Address High 7 (ERSTBA_HI6)—Offset 20F4h.....1317
16.158Event Ring Dequeue Pointer Low 7 (ERDP_LO6)—Offset 20F8h 1318
16.159Event Ring Dequeue Pointer High 7 (ERDP_HI6)—Offset 20FCh.....1318
16.160Interrupter 8 Management (IMAN7)—Offset 2100h.....1318
16.161Interrupter 8 Moderation (IMOD7)—Offset 2104h.....1319
16.162Event Ring Segment Table Size 8 (ERSTSZ7)—Offset 2108h.....1319
16.163Event Ring Segment Table Base Address Low 8 (ERSTBA_LO7)—Offset 2110h 1320
16.164Event Ring Segment Table Base Address High 8 (ERSTBA_HI7)—Offset 2114h.....1320
16.165Event Ring Dequeue Pointer Low 8 (ERDP_LO7)—Offset 2118h 1320
16.166Event Ring Dequeue Pointer High 8 (ERDP_HI7)—Offset 211Ch.....1321
16.167Door Bell 1 (DB0)—Offset 3000h.....1321
16.168Door Bell 2 (DB1)—Offset 3004h.....1321
16.169Door Bell 3 (DB2)—Offset 3008h.....1322
16.170Door Bell 4 (DB3)—Offset 300Ch 1322
16.171Door Bell 5 (DB4)—Offset 3010h.....1323
16.172Door Bell 6 (DB5)—Offset 3014h.....1323
16.173Door Bell 7 (DB6)—Offset 3018h.....1324
16.174Door Bell 8 (DB7)—Offset 301Ch 1324
16.175Door Bell 9 (DB8)—Offset 3020h.....1324
16.176Door Bell 10 (DB9)—Offset 3024h.....1325
16.177Door Bell 11 (DB10)—Offset 3028h 1325
16.178Door Bell 12 (DB11)—Offset 302Ch.....1326
16.179Door Bell 13 (DB12)—Offset 3030h 1326
16.180Door Bell 14 (DB13)—Offset 3034h 1326
16.181Door Bell 15 (DB14)—Offset 3038h 1327
16.182Door Bell 16 (DB15)—Offset 303Ch.....1327
16.183Door Bell 17 (DB16)—Offset 3040h 1328
16.184Door Bell 18 (DB17)—Offset 3044h 1328
16.185Door Bell 19 (DB18)—Offset 3048h 1328
16.186Door Bell 20 (DB19)—Offset 304Ch.....1329
16.187Door Bell 21 (DB20)—Offset 3050h 1329
16.188Door Bell 22 (DB21)—Offset 3054h 1330
16.189Door Bell 23 (DB22)—Offset 3058h 1330
16.190Door Bell 24 (DB23)—Offset 305Ch.....1330
16.191Door Bell 25 (DB24)—Offset 3060h 1331
16.192Door Bell 26 (DB25)—Offset 3064h 1331
16.193Door Bell 27 (DB26)—Offset 3068h 1332
16.194Door Bell 28 (DB27)—Offset 306Ch.....1332
16.195Door Bell 29 (DB28)—Offset 3070h 1332
16.196Door Bell 30 (DB29)—Offset 3074h 1333
16.197Door Bell 31 (DB30)—Offset 3078h 1333
16.198Door Bell 32 (DB31)—Offset 307Ch.....1334
16.199Door Bell 32 (DB32)—Offset 3080h 1334

16.200XECP_SUPP_USB2_0 (XECP_SUPP_USB2_0)—Offset 8000h 1334
16.201XECP_SUPP_USB2_1 (XECP_SUPP_USB2_1)—Offset 8004h 1335
16.202XECP_SUPP_USB2_2 (XECP_SUPP_USB2_2)—Offset 8008h 1335
16.203XECP_SUPP_USB2_3 (Full Speed) (XECP_SUPP_USB2_3)—Offset 8010h 1336
16.204XECP_SUPP_USB2_4 (Low Speed) (XECP_SUPP_USB2_4)—Offset 8014h..... 1336
16.205XECP_SUPP_USB2_5 (High Speed) (XECP_SUPP_USB2_5)—Offset 8018h..... 1337
16.206XECP_SUPP_USB3_0 (XECP_SUPP_USB3_0)—Offset 8020h 1337
16.207XECP_SUPP_USB3_1 (XECP_SUPP_USB3_1)—Offset 8024h 1338
16.208XECP_SUPP_USB3_2 (XECP_SUPP_USB3_2)—Offset 8028h 1338
16.209XECP_SUPP_USB3_3 (XECP_SUPP_USB3_3)—Offset 8030h 1339
16.210XECP_SUPP_USB3_4 (XECP_SUPP_USB3_4)—Offset 8034h 1339
16.211XECP_SUPP_USB3_5 (XECP_SUPP_USB3_5)—Offset 8038h 1340
16.212XECP_SUPP_USB3_6 (XECP_SUPP_USB3_6)—Offset 803Ch 1340
16.213XECP_SUPP_USB3_7 (XECP_SUPP_USB3_7)—Offset 8040h 1341
16.214XECP_SUPP_USB3_8 (XECP_SUPP_USB3_8)—Offset 8044h 1341
16.215XECP_SUPP_USB3_9 (XECP_SUPP_USB3_9)—Offset 8048h 1342
16.216Host Controller Capability (HOST_CTRL_CAP_REG)—Offset 8070h 1342
16.217Override EP Flow Control (HOST_CLR_MASK_REG)—Offset 8078h 1343
16.218Clear Active IN EP ID Control (HOST_CLR_IN_EP_VALID_REG)—Offset 807Ch..... 1343
16.219Clear Poll Mask Control (HOST_CLR_PMASK_REG)—Offset 8080h 1343
16.220Host Control Scheduler (HOST_CTRL_SCH_REG)—Offset 8094h 1344
16.221Global Port Control (HOST_CTRL_PORT_CTRL)—Offset 80A0h 1345
16.222Power Management Control (PMCTRL_REG)—Offset 80A4h 1345
16.223PGCB Control (PGCBCTRL_REG)—Offset 80A8h 1347
16.224D0I3 Control (DOI3CTRL_REG)—Offset 80ACh 1350
16.225HOST_CTRL_MISC_REG (HOST_CTRL_MISC_REG)—Offset 80B0h..... 1351
16.226HOST_CTRL_MISC_REG2 (HOST_CTRL_MISC_REG2)—Offset 80B4h 1352
16.227SSPE_REG (SSPE_REG)—Offset 80B8h..... 1353
16.228(SSPITPE)—Offset 80BCh 1353
16.229AUX Reset Control (AUX_CTRL_REG)—Offset 80C0h 1354
16.230Super Speed Bandwidth Overload (HOST_BW_OV_SS_REG)—Offset 80C4h 1356
16.231High Speed TT Bandwidth Overload (HOST_BW_OV_HS_REG)—Offset 80C8h..... 1356
16.232Bandwidth Overload Full Low Speed (HOST_BW_OV_FS_LS_REG)—Offset 80CCh .. 1357
16.233System Bandwidth Overload (HOST_BW_OV_SYS_REG)—Offset 80D0h..... 1357
16.234Scheduler Async Delay (HOST_CTRL_SCH_ASYNC_DELAY_REG)—Offset 80D4h..... 1358
16.235DEVICE MODE CONTROL REG 0 (DUAL_ROLE_CFG_REG0)—Offset 80D8h..... 1358
16.236DEVICE MODE CONTROL REG 1 (DUAL_ROLE_CFG_REG1)—Offset 80DCh..... 1360
16.237AUX Power Management Control (AUX_CTRL_REG1)—Offset 80E0h 1361
16.238Battery Charge (BATTERY_CHARGE_REG)—Offset 80E4h..... 1363
16.239Port Watermark (HOST_CTRL_WATERMARK_REG)—Offset 80E8h 1363
16.240SuperSpeed Port Link Control (HOST_CTRL_PORT_LINK_REG)—Offset 80ECh 1364
16.241USB2 Port Link Control 1 (USB2_LINK_MGR_CTRL_REG1)—Offset 80F0h..... 1365
16.242USB2 Port Link Control 2 (USB2_LINK_MGR_CTRL_REG2)—Offset 80F4h..... 1367
16.243USB2 Port Link Control 3 (USB2_LINK_MGR_CTRL_REG3)—Offset 80F8h..... 1367
16.244USB2 Port Link Control 4 (USB2_LINK_MGR_CTRL_REG4)—Offset 80FCh..... 1368
16.245Bandwidth Calc Control (HOST_CTRL_BW_CTRL_REG)—Offset 8100h..... 1368
16.246Host Interface Control (HOST_IF_CTRL_REG)—Offset 8108h..... 1368
16.247Bandwidth Overload Burst (HOST_BW_OV_BURST_REG)—Offset 810Ch 1369
16.248USB Max Bandwidth Control 4 (HOST_CTRL_BW_MAX_REG)—Offset 8128h 1369
16.249USB2 Linestate Debug (LINESTATE_DEBUG_REG)—Offset 8130h..... 1370
16.250USB2 Protocol Gap Timer (USB2_PROTOCOL_GAP_TIMER_REG)—Offset 8134h 1371
16.251USB2 Protocol Bus Timeout Timer (USB2_PROTOCOL_BTO_TIMER_REG)—Offset 813Ch 1371
16.252Power Scheduler Control-0 (PWR_SCHED_CTRL0)—Offset 8140h..... 1372
16.253Power Scheduler Control-2 (PWR_SCHED_CTRL2)—Offset 8144h..... 1372

16.254AUX Power Management Control (AUX_CTRL_REG2)—Offset 8154h 1373
16.255USB2 PHY Power Management Control (USB2_PHY_PMC)—Offset 8164h 1375
16.256USB Power Gating Control (USB_PGC)—Offset 8168h 1376
16.257xHCI Aux Clock Control Register (XHCI_AUX_CCR)—Offset 816Ch.....1376
16.258USB LPM Parameters (USB_LPM_PARAM)—Offset 8170h 1379
16.259xHC Latency Tolerance Parameters - LTV Control (XLTP_LTV1)—Offset 8174h.....1379
16.260xHC Latency Tolerance Parameters LTV Control 2 (XLTP_LTV2)—Offset 8178h 1380
16.261xHC Latency Tolerance Parameters - High Idle Time Control (XLTP_HITC)—Offset 817Ch 1381
16.262xHC Latency Tolerance Parameters - Medium Idle Time Control (XLTP_MITC)—Offset 8180h.....1381
16.263xHC Latency Tolerance Parameters Low Idle Time Control (XLTP_LITC)—Offset 8184h ... 1382
16.264HOST_CTRL_BW_MAX3_REG (HOST_CTRL_BW_MAX3_REG)—Offset 8188h.....1382
16.265PDDIS_REG (PDDIS_REG)—Offset 8198h.....1383
16.266THRM_HOST_CTRL_REG (THRM_HOST_CTRL_REG)—Offset 819Ch.....1383
16.267LFPS_PM_CTRL_REG (LFPS_PM_CTRL_REG)—Offset 81A0h 1385
16.268U2PDM (U2PDM)—Offset 81A4h.....1385
16.269U2PCM (U2PCM)—Offset 81A8h 1385
16.270U3PDM (U3PDM)—Offset 81ACh.....1386
16.271U3PCM (U3PCM)—Offset 81B0h 1386
16.272THRM_HOST_CTRL_REG2 (THRM_HOST_CTRL_REG2)—Offset 81B4h 1386
16.273LFPSONCOUNT_REG (LFPSONCOUNT_REG)—Offset 81B8h 1387
16.274(USB2PMCTRL_REG)—Offset 81C4h 1387
16.275ECC_PARITY_ERROR_LOG_REG (ECC_PARITY_ERROR_LOG_REG)—Offset 83F8h ...1388
16.276ECC_POISONING_CTRL_REG (ECC_POISONING_CTRL_REG)—Offset 83FCh 1390
16.277USB2_PORT_STATE_REG (USB2_PORT_STATE_REG)—Offset 8400h 1391
16.278USB3_PORT_STATE_REG (USB3_PORT_STATE_REG)—Offset 8408h 1391
16.279FUS1_REG (FUS1_REG)—Offset 8410h 1392
16.280FUS2_REG (FUS2_REG)—Offset 8414h 1393
16.281FUS3_REG (FUS3_REG)—Offset 8418h 1393
16.282STRAP1_REG (STRAP1_REG)—Offset 841Ch 1393
16.283STRAP2_REG (STRAP2_REG)—Offset 8420h 1394
16.284STRAP3_REG (STRAP3_REG)—Offset 8424h 1394
16.285DFT_REG1 (DFT_REG1)—Offset 8430h 1395
16.286DFT_REG2 (DFT_REG2)—Offset 8434h 1395
16.287DFT_REG3 (DFT_REG3)—Offset 8438h 1396
16.288dft_reg4 (DFT_REG4)—Offset 843Ch 1397
16.289dft_reg5 (DFT_REG5)—Offset 8440h 1397
16.290XECP_CMDM_STS0 (XECP_CMDM_STS0)—Offset 8448h 1398
16.291XECP_CMDM_STS1 (XECP_CMDM_STS1)—Offset 844Ch 1399
16.292XECP_CMDM_STS2 (XECP_CMDM_STS2)—Offset 8450h 1400
16.293XECP_CMDM_STS3 (XECP_CMDM_STS3)—Offset 8454h 1400
16.294XECP_CMDM_STS4 (XECP_CMDM_STS4)—Offset 8458h 1400
16.295XECP_CMDM_STS5 (XECP_CMDM_STS5)—Offset 845Ch 1400
16.296AUX Power PHY Reset (UPORTS_PON_RST_REG)—Offset 8460h.....1401
16.297Latency Tolerance Control 0 (HOST_IF_LAT_TOL_CTRL_REG0)—Offset 8464h.....1401
16.298USB Legacy Support Capability (USBLEGSUP)—Offset 846Ch 1402
16.299USB Legacy Support Control Status (USBLEGCTLSTS)—Offset 8470h.....1403
16.300Port Disable Override capability register (PDO_CAPABILITY)—Offset 84F4h.....1403
16.301USB2 Port Disable Override (USB2PDO)—Offset 84F8h 1404
16.302USB3 Port Disable Override (USB3PDO)—Offset 84FCh 1404
16.303HW state capability register (HW_STATE_CAPABILITY)—Offset 8500h.....1404
16.304HW state register 1 (HW_STATE_REG1)—Offset 8504h 1405
16.305HW state register 2 (HW_STATE_REG2)—Offset 8508h 1405

16.306HW state register 3 (HW_STATE_REG3)—Offset 850Ch 1405
16.307HW state register 4 (HW_STATE_REG4)—Offset 8510h 1406
16.308CONFIG mirror capability register (CONFIG_MIRROR_CAPABILITY)—Offset 8600h.. 1406
16.309Command (CMD_MMIO)—Offset 8604h 1406
16.310Device Status (STS_MMIO)—Offset 8606h 1407
16.311Revision ID (RID_MMIO)—Offset 8608h 1408
16.312Programming Interface (PI_MMIO)—Offset 8609h..... 1409
16.313Sub Class Code (SCC_MMIO)—Offset 860Ah 1409
16.314Base Class Code (BCC_MMIO)—Offset 860Bh 1409
16.315Master Latency Timer (MLT_MMIO)—Offset 860Dh..... 1410
16.316Header Type (HT_MMIO)—Offset 860Eh 1410
16.317Memory Base Address (MBAR_MMIO)—Offset 8610h..... 1410
16.318USB Subsystem Vendor ID (SSVID_MMIO)—Offset 862Ch 1411
16.319USB Subsystem ID (SSID_MMIO)—Offset 862Eh 1411
16.320Capabilities Pointer (CAP_PTR_MMIO)—Offset 8634h 1412
16.321Interrupt Line (ILINE_MMIO)—Offset 863Ch 1412
16.322Interrupt Pin (IPIN_MMIO)—Offset 863Dh 1412
16.323XHC System Bus Configuration 1 (XHCC1_MMIO)—Offset 8640h 1413
16.324XHC System Bus Configuration 2 (XHCC2_MMIO)—Offset 8644h 1414
16.325Clock Gating (XHCLKGTEN_MMIO)—Offset 8650h 1416
16.326Audio Time Synchronization (AUDSYNC_MMIO)—Offset 8658h 1419
16.327Serial Bus Release Number (SBRN_MMIO)—Offset 8660h 1420
16.328Frame Length Adjustment (FLADJ_MMIO)—Offset 8661h 1420
16.329Best Effort Service Latency (BESL_MMIO)—Offset 8662h 1421
16.330PCI Power Management Capability ID (PM_CID_MMIO)—Offset 8670h..... 1421
16.331Next Item Pointer #1 (PM_NEXT_MMIO)—Offset 8671h..... 1421
16.332Power Management Capabilities (PM_CAP_MMIO)—Offset 8672h 1422
16.333Power Management Control/Status (PM_CS_MMIO)—Offset 8674h..... 1423
16.334Message Signaled Interrupt CID (MSI_CID_MMIO)—Offset 8680h..... 1424
16.335Next item pointer (MSI_NEXT_MMIO)—Offset 8681h..... 1424
16.336Message Signaled Interrupt Message Control (MSI_MCTL_MMIO)—Offset 8682h 1424
16.337Message Signaled Interrupt Message Address (MSI_MAD_MMIO)—Offset 8684h 1425
16.338Message Signaled Interrupt Upper Address (MSI_MUAD_MMIO)—Offset 8688h..... 1425
16.339Message Signaled Interrupt Message Data (MSI_MD_MMIO)—Offset 868Ch 1426
16.340Device Idle Capability (DEVIDLE_MMIO)—Offset 8690h 1426
16.341Vendor Specific Header (VSHDR_MMIO)—Offset 8694h 1427
16.342SW LTR POINTER (SWLTRPTR_MMIO)—Offset 8698h 1427
16.343Device Idle Pointer Register (DEVIDLEPTR_MMIO)—Offset 869Ch 1428
16.344Device Idle Power ON Latency (DEVIDLEPOL_MMIO)—Offset 86A0h 1429
16.345High Speed Configuration 2 (HSCFG2_MMIO)—Offset 86A4h 1429
16.346XHCI USB2 Overcurrent Pin Mapping 1 (U2OCM1_MMIO)—Offset 86B0h 1431
16.347XHCI USB2 Overcurrent Pin Mapping 2 (U2OCM2_MMIO)—Offset 86B4h 1431
16.348XHCI USB3 Overcurrent Pin Mapping 1 (U3OCM1_MMIO)—Offset 86D0h..... 1431
16.349XHCI USB3 Overcurrent Pin Mapping 2 (U3OCM2_MMIO)—Offset 86D4h..... 1432
16.350XHCC3 (XHCC3_MMIO)—Offset 86FCh 1432
16.351Debug Capability ID Register (DCID)—Offset 8700h..... 1433
16.352Debug Capability Doorbell Register (DCDB)—Offset 8704h..... 1433
16.353Debug Capability Event Ring Segment Table Size Register (DCERSTSZ)—Offset 8708h .. 1434
16.354Debug Capability Event Ring Segment Table Base Address Register (DCERSTBA)—Offset 8710h 1434
16.355Debug Capability Event Ring Dequeue Pointer Register (DCERDP)—Offset 8718h ... 1435
16.356Debug Capability Control Register (DCCTRL)—Offset 8720h 1435
16.357Debug Capability Status Register (DCST)—Offset 8724h..... 1436
16.358Debug Capability Port Status and Control Register (DCPORTSC)—Offset 8728h..... 1436

16.359Debug Capability Context Pointer Register (DCCP)—Offset 8730h 1438
16.360Debug Capability Device Descriptor Info Register 1 (DCDDI1)—Offset 8738h 1439
16.361Debug Capability Device Descriptor Info Register 2 (DCDDI2)—Offset 873Ch 1439
16.362Debug Capability Descriptor Parameters (DCDP)—Offset 8740h 1439
16.363Debug Device Control ODMA (DBGDEV_CTRL_ODMA_REG)—Offset 8748h.....1440
16.364DBC Control Register 1 (DBCCTL_REG)—Offset 8760h 1441
16.365SSIC Policy and Implementation Specific Registers Capability ID register (SSIC_IMPLEMENTATION_CAPABILITY_ID_REG)—Offset 8800h 1442
16.366SSIC global configuration control register (SSIC_GLOBAL_CONFIG_CONTROL_REG)—Offset 8804h 1442
16.367 (PORT1_SSIC_CONFIG_REG1)—Offset 8808h 1443
16.368 (PORT1_SSIC_CONFIG_REG2)—Offset 880Ch 1444
16.369 (PORT1_SSIC_CONFIG_REG3)—Offset 8810h 1445
16.370 (PORT1_SSIC_CONFIG_REG4)—Offset 8814h 1446
16.371 (PORT1_SSIC_LOOPBACK_CONFIG_REG)—Offset 8818h.....1446
16.372 (PORT1_SSIC_LOOPBACK_BURST_COUNT_REG)—Offset 881Ch.....1446
16.373 (PORT1_SSIC_LOOPBACK_ERROR_COUNT_REG)—Offset 8820h.....1447
16.374SSIC Local and Remote Profile Registers Capability ID register (SSIC_PROFILE_CAPABILITY_ID_REG)—Offset 8900h 1448
16.375SSIC Port N Register Access Control (PORT1_REGISTER_ACCESS_CONTROL)—Offset 8904h.....1448
16.376SSIC Port N Register Access Status (PORT1_REGISTER_ACCESS_STATUS)—Offset 8908h 1449
16.377 (PORT1_PROFILE_ATTRIBUTES_REG0)—Offset 890Ch 1450
16.378(PORT1_PROFILE_ATTRIBUTES_REG1)—Offset 8910h 1450
16.379(PORT1_PROFILE_ATTRIBUTES_REG2)—Offset 8914h 1451
16.380(PORT1_PROFILE_ATTRIBUTES_REG3)—Offset 8918h 1451
16.381(PORT1_PROFILE_ATTRIBUTES_REG4)—Offset 891Ch 1452
16.382(PORT1_PROFILE_ATTRIBUTES_REG5)—Offset 8920h 1452
16.383(PORT1_PROFILE_ATTRIBUTES_REG6)—Offset 8924h 1453
16.384(PORT1_PROFILE_ATTRIBUTES_REG7)—Offset 8928h 1453
16.385(PORT1_PROFILE_ATTRIBUTES_REG8)—Offset 892Ch 1454
16.386(PORT1_PROFILE_ATTRIBUTES_REG9)—Offset 8930h 1454
16.387(PORT1_PROFILE_ATTRIBUTES_REG10)—Offset 8934h.....1455
16.388(PORT1_PROFILE_ATTRIBUTES_REG11)—Offset 8938h.....1455
16.389(PORT1_PROFILE_ATTRIBUTES_REG12)—Offset 893Ch 1456
16.390(PORT1_PROFILE_ATTRIBUTES_REG13)—Offset 8940h.....1456
16.391(PORT1_PROFILE_ATTRIBUTES_REG14)—Offset 8944h.....1457
16.392(PORT1_PROFILE_ATTRIBUTES_REG15)—Offset 8948h.....1457
16.393(PORT1_PROFILE_ATTRIBUTES_REG16)—Offset 894Ch 1458
16.394(PORT1_PROFILE_ATTRIBUTES_REG17)—Offset 8950h.....1458
16.395(PORT1_PROFILE_ATTRIBUTES_REG18)—Offset 8954h.....1459
16.396(PORT1_PROFILE_ATTRIBUTES_REG19)—Offset 8958h.....1459
16.397(PORT1_PROFILE_ATTRIBUTES_REG20)—Offset 895Ch 1460
16.398(PORT1_PROFILE_ATTRIBUTES_REG21)—Offset 8960h.....1460
16.399(PORT1_PROFILE_ATTRIBUTES_REG22)—Offset 8964h.....1461
16.400(PORT1_PROFILE_ATTRIBUTES_REG23)—Offset 8968h.....1461
16.401(PORT1_PROFILE_ATTRIBUTES_REG24)—Offset 896Ch 1462
16.402(PORT1_PROFILE_ATTRIBUTES_REG25)—Offset 8970h.....1462
16.403(PORT1_PROFILE_ATTRIBUTES_REG26)—Offset 8974h.....1463
16.404(PORT1_PROFILE_ATTRIBUTES_REG27)—Offset 8978h.....1463
16.405(PORT1_PROFILE_ATTRIBUTES_REG28)—Offset 897Ch 1464
16.406(PORT1_PROFILE_ATTRIBUTES_REG29)—Offset 8980h.....1464
16.407(PORT1_PROFILE_ATTRIBUTES_REG30)—Offset 8984h.....1465
16.408(PORT1_PROFILE_ATTRIBUTES_REG31)—Offset 8988h.....1465

16.409(PORT1_PROFILE_ATTRIBUTES_REG32)—Offset 898Ch 1466
16.410(PORT1_PROFILE_ATTRIBUTES_REG33)—Offset 8990h 1466
16.411(PORT1_PROFILE_ATTRIBUTES_REG34)—Offset 8994h 1467
16.412(PORT1_PROFILE_ATTRIBUTES_REG35)—Offset 8998h 1467
16.413(PORT1_PROFILE_ATTRIBUTES_REG36)—Offset 899Ch 1468
16.414(PORT1_PROFILE_ATTRIBUTES_REG37)—Offset 89A0h 1468
16.415(PORT1_PROFILE_ATTRIBUTES_REG38)—Offset 89A4h 1469
16.416(PORT1_PROFILE_ATTRIBUTES_REG39)—Offset 89A8h 1469
16.417(PORT1_PROFILE_ATTRIBUTES_REG40)—Offset 89ACh 1470
16.418(PORT1_PROFILE_ATTRIBUTES_REG41)—Offset 89B0h 1470
16.419(PORT1_PROFILE_ATTRIBUTES_REG42)—Offset 89B4h 1471
16.420(PORT1_PROFILE_ATTRIBUTES_REG43)—Offset 89B8h 1471
16.421(PORT1_PROFILE_ATTRIBUTES_REG44)—Offset 89BCh 1472
16.422(PORT1_PROFILE_ATTRIBUTES_REG45)—Offset 89C0h 1472
16.423(PORT1_PROFILE_ATTRIBUTES_REG46)—Offset 89C4h 1473
16.424(PORT1_PROFILE_ATTRIBUTES_REG47)—Offset 89C8h 1473
16.425(PORT1_PROFILE_ATTRIBUTES_REG48)—Offset 89CCh 1474
16.426(PORT1_PROFILE_ATTRIBUTES_REG49)—Offset 89D0h 1474
16.427(PORT1_PROFILE_ATTRIBUTES_REG50)—Offset 89D4h 1475
16.428(PORT1_PROFILE_ATTRIBUTES_REG51)—Offset 89D8h 1475
16.429(PORT1_PROFILE_ATTRIBUTES_REG52)—Offset 89DCh 1476
16.430(PORT1_PROFILE_ATTRIBUTES_REG53)—Offset 89E0h 1476
16.431(PORT1_PROFILE_ATTRIBUTES_REG54)—Offset 89E4h 1477
16.432(PORT1_PROFILE_ATTRIBUTES_REG55)—Offset 89E8h 1477
16.433(PORT1_PROFILE_ATTRIBUTES_REG56)—Offset 89ECh 1478
16.434(PORT1_PROFILE_ATTRIBUTES_REG57)—Offset 89F0h 1478
16.435(PORT1_PROFILE_ATTRIBUTES_REG58)—Offset 89F4h 1479
16.436(PORT1_PROFILE_ATTRIBUTES_REG59)—Offset 89F8h 1479
16.437(PORT1_PROFILE_ATTRIBUTES_REG60)—Offset 89FCh 1480
16.438(PORT1_PROFILE_ATTRIBUTES_REG61)—Offset 8A00h 1480
16.439(PORT1_PROFILE_ATTRIBUTES_REG62)—Offset 8A04h 1481
16.440(PORT1_PROFILE_ATTRIBUTES_REG63)—Offset 8A08h 1481
16.441GLOBAL_TIME_SYNC_CAP_REG (GLOBAL_TIME_SYNC_CAP_REG)—Offset 8E10h... 1482
16.442GLOBAL_TIME_SYNC_CTRL_REG (GLOBAL_TIME_SYNC_CTRL_REG)—Offset 8E14h 1482
16.443MICROFRAME_TIME_REG (MICROFRAME_TIME_REG)—Offset 8E18h 1482
16.444GLOBAL_TIME_LOW_REG (GLOBAL_TIME_LOW_REG)—Offset 8E20h 1483
16.445GLOBAL_TIME_HI_REG (GLOBAL_TIME_HI_REG)—Offset 8E24h..... 1483
16.446Debug Status Capability Register (DEBUG_STATUS_CAPABILITY_REG)—Offset 8E58h ... 1484
16.447Host Ctrl USB3 Soft Error Count Register 1 (HOST_CTRL_USB3_ERR_COUNT_REG1)—Offset 8E5Ch..... 1484
16.448Host Ctrl USB3 Soft Error Count Register 2 (HOST_CTRL_USB3_ERR_COUNT_REG2)—Offset 8E60h..... 1484
16.449Host Ctrl USB3 Soft Error Count Register 3 (HOST_CTRL_USB3_ERR_COUNT_REG3)—Offset 8E64h..... 1485
16.450Host Ctrl USB3 Soft Error Count Register 4 (HOST_CTRL_USB3_ERR_COUNT_REG4)—Offset 8E68h..... 1485
16.451Host Ctrl USB3 Soft Error Count Register 5 (HOST_CTRL_USB3_ERR_COUNT_REG5)—Offset 8E6Ch..... 1486
16.452Host Ctrl USB3 Soft Error Count Register 6 (HOST_CTRL_USB3_ERR_COUNT_REG6)—Offset 8E70h..... 1486
16.453Host Ctrl USB3 Soft Error Count Register 7 (HOST_CTRL_USB3_ERR_COUNT_REG7)—Offset 8E74h..... 1486
16.454DBC_GP2_OUT_PAYLOAD_BP_LOW (DBC_GP2_OUT_PAYLOAD_BP_LOW)—Offset 0h 1487

16.455DBC_GP2_OUT_PAYLOAD_BP_HI (DBC_GP2_OUT_PAYLOAD_BP_HI)—Offset 4h 1487
16.456DBC_GP2_OUT_PAYLOAD_QUALIFIERS (DBC_GP2_OUT_PAYLOAD_QUALIFIERS)—Offset 8h....1487
16.457DBC_GP2_OUT_PAYLOAD_TRANSFER_LENGTH (DBC_GP2_OUT_PAYLOAD_TRANSFER_LENGTH)—Offset Ch 1488
16.458DBC_GP2_OUT_STATUS_BP_LOW (DBC_GP2_OUT_STATUS_BP_LOW)—Offset 10h 1488
16.459DBC_GP2_OUT_STATUS_BP_HI (DBC_GP2_OUT_STATUS_BP_HI)—Offset 14h 1489
16.460DBC_GP2_OUT_STATUS_QUALIFIERS (DBC_GP2_OUT_STATUS_QUALIFIERS)—Offset 18h 1489
16.461DBC_GP2_IN_PAYLOAD_BP_LOW (DBC_GP2_IN_PAYLOAD_BP_LOW)—Offset 1Ch .1490
16.462DBC_GP2_IN_PAYLOAD_BP_HI (DBC_GP2_IN_PAYLOAD_BP_HI)—Offset 20h 1490
16.463DBC_GP2_IN_PAYLOAD_QUALIFIERS (DBC_GP2_IN_PAYLOAD_QUALIFIERS)—Offset 24h 1490
16.464DBC_GP2_IN_PAYLOAD_TRANSFER_LENGTH (DBC_GP2_IN_PAYLOAD_TRANSFER_LENGTH)—Offset 28h....1491
16.465DBC_GP2_IN_STATUS_BP_LOW (DBC_GP2_IN_STATUS_BP_LOW)—Offset 2Ch 1491
16.466DBC_GP2_IN_STATUS_BP_HI (DBC_GP2_IN_STATUS_BP_HI)—Offset 30h.....1492
16.467DBC_GP2_IN_STATUS_QUALIFIERS (DBC_GP2_IN_STATUS_QUALIFIERS)—Offset 34h.. 1492
16.468DBC_DFX_OUT_CONTROL (DBC_DFX_OUT_CONTROL)—Offset 38h 1492
16.469DBC_DFX_IN_PAYLOAD_BP_LOW (DBC_DFX_IN_PAYLOAD_BP_LOW)—Offset 3Ch .1493
16.470DBC_DFX_IN_PAYLOAD_BP_HI (DBC_DFX_IN_PAYLOAD_BP_HI)—Offset 40h.....1493
16.471DBC_DFX_IN_PAYLOAD_TRANSFER_LENGTH (DBC_DFX_IN_PAYLOAD_TRANSFER_LENGTH)—Offset 44h....1494
16.472DBC_DFX_IN_STATUS_BP_LOW (DBC_DFX_IN_STATUS_BP_LOW)—Offset 48h 1494
16.473DBC_DFX_IN_STATUS_BP_HI (DBC_DFX_IN_STATUS_BP_HI)—Offset 4Ch 1494
16.474DBC_TRACE_IN_PAYLOAD_BP_LOW (DBC_TRACE_IN_PAYLOAD_BP_LOW)—Offset 50h . 1495
16.475DBC_TRACE_IN_PAYLOAD_BP_HI (DBC_TRACE_IN_PAYLOAD_BP_HI)—Offset 54h.1495
16.476DBC_TRACE_IN_PAYLOAD_QUALIFIERS (DBC_TRACE_IN_PAYLOAD_QUALIFIERS)—Offset 58h 1496
16.477DBC_TRACE_IN_PAYLOAD_TRASNFER_DOORBELL (DBC_TRACE_IN_PAYLOAD_TRANSFER_DOORBELL)—Offset 5Ch....1496
16.478DBC_TRACE_IN_STATUS_BP_LOW (DBC_TRACE_IN_STATUS_BP_LOW)—Offset 60h 1496
16.479DBC_TRACE_IN_STATUS_BP_HI (DBC_TRACE_IN_STATUS_BP_HI)—Offset 64h.....1497
16.480DBC_TRACE_IN_STATUS_QUALIFIERS (DBC_TRACE_IN_STATUS_QUALIFIERS)—Offset 68h 1497
16.481DBC_ERROR_CONTROL_STATUS_REG (DBC_ERROR_CONTROL_STATUS_REG)—Offset 6Ch 1498
16.482DBC_EXI_CONTROL_STATUS_REG (DBC_EXI_CONTROL_STATUS_REG)—Offset 70h 1498
16.483FABRIC_AGENT_UPSTREAM_GRANT_COUNT_REG (FABRIC_AGENT_UPSTREAM_GRANT_COUNT_REG)—Offset 74h 1499
16.484DBC_ECO_POLICY_REG1 (DBC_ECO_POLICY_REG1)—Offset 78h 1500
16.485DBC_ECO_POLICY_REG2 (DBC_ECO_POLICY_REG2)—Offset 7Ch.....1502
16.486DBC_ECO_POLICY_REG3 (DBC_ECO_POLICY_REG3)—Offset 80h 1503
16.487DBC_ECO_POLICY_REG4 (DBC_ECO_POLICY_REG4)—Offset 84h 1503
16.488DBConEXI Capability Port Status and Control Register (DBC_EXI_DCPORTSC)—Offset 88h 1504
16.489DBC_GP2_OUT_DMA_STS_REG1 (DBC_GP2_OUT_DMA_STS_REG1)—Offset 8Ch ...1505
16.490DBC_GP2_OUT_DMA_STS_REG2 (DBC_GP2_OUT_DMA_STS_REG2)—Offset 90h ...1506
16.491DBC_GP2_IN_DMA_STS_REG1 (DBC_GP2_IN_DMA_STS_REG1)—Offset 94h.....1506
16.492DBC_GP2_IN_DMA_STS_REG2 (DBC_GP2_IN_DMA_STS_REG2)—Offset 98h.....1507
16.493DEBUG_SW_CONTROL_STATUS_REG (DEBUG_SW_CONTROL_STATUS_REG)—Offset 100h 1507

16.494DEBUG_REQUEST_INFO_AND_STATUS_REG (DEBUG_REQUEST_INFO_AND_STATUS_REG)—Offset 104h 1508
16.495DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_0)—Offset 108h 1509
16.496DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_1)—Offset 10Ch 1509
16.497DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_2)—Offset 110h 1510
16.498DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_3)—Offset 114h 1510
16.499DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_4)—Offset 118h 1510
16.500DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_5)—Offset 11Ch 1511
16.501DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_6)—Offset 120h 1511
16.502DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_7)—Offset 124h 1511
16.503DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_8)—Offset 128h 1512
16.504DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_9)—Offset 12Ch 1512
16.505DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_10)—Offset 130h 1512
16.506DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_11)—Offset 134h 1513
16.507DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_12)—Offset 138h 1513
16.508DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_13)—Offset 13Ch 1513
16.509DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_14)—Offset 140h 1514
16.510DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_15)—Offset 144h 1514
16.511DEBUG_RESPONSE_INFO_AND_STATUS_REG (DEBUG_RESPONSE_INFO_AND_STATUS_REG)—Offset 148h 1514
16.512DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_0)—Offset 180h 1515
16.513DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_1)—Offset 184h 1515
16.514DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_2)—Offset 188h 1515
16.515DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_3)—Offset 18Ch 1516
16.516DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_4)—Offset 190h 1516
16.517DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_5)—Offset 194h 1516
16.518DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_6)—Offset 198h 1517
16.519DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_7)—Offset 19Ch 1517
16.520DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_8)—Offset 1A0h 1517
16.521DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_9)—Offset 1A4h 1518
16.522DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_10)—Offset 1A8h 1518
16.523DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_11)—Offset 1ACh 1518
16.524DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_12)—Offset 1B0h 1519
16.525DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_13)—Offset 1B4h 1519
16.526DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_14)—Offset 1B8h 1519
16.527DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_15)—Offset 1BCh 1520
16.528DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_16)—Offset 1C0h 1520
16.529DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_17)—Offset 1C4h 1520

16.530DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_18)—Offset 1C8h 1521
16.531DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_19)—Offset 1CCh 1521
16.532DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_20)—Offset 1D0h 1521
16.533DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_21)—Offset 1D4h 1522
16.534DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_22)—Offset 1D8h 1522
16.535DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_23)—Offset 1DCh 1522
16.536DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_24)—Offset 1E0h 1523
16.537DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_25)—Offset 1E4h 1523
16.538DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_26)—Offset 1E8h 1523
16.539DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_27)—Offset 1ECh 1524
16.540DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_28)—Offset 1F0h.....1524
16.541DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_29)—Offset 1F4h.....1524
16.542DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_30)—Offset 1F8h.....1525
16.543DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_31)—Offset 1FCh 1525
16.544DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_32)—Offset 200h 1525
16.545DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_33)—Offset 204h 1526
16.546DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_34)—Offset 208h 1526
16.547DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_35)—Offset 20Ch 1526
16.548DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_36)—Offset 210h 1527
16.549DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_37)—Offset 214h 1527
16.550DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_38)—Offset 218h 1527
16.551DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_39)—Offset 21Ch 1528
16.552DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_40)—Offset 220h 1528
16.553DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_41)—Offset 224h 1528
16.554DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_42)—Offset 228h 1529
16.555DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_43)—Offset 22Ch 1529
16.556DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_44)—Offset 230h 1529
16.557DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_45)—Offset 234h 1530
16.558DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_46)—Offset

238h 1530
16.559DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_47)—Offset 23Ch 1530
16.560DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_48)—Offset 240h 1531
16.561DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_49)—Offset 244h 1531
16.562DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_50)—Offset 248h 1531
16.563DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_51)—Offset 24Ch 1532
16.564DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_52)—Offset 250h 1532
16.565DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_53)—Offset 254h 1532
16.566DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_54)—Offset 258h 1533
16.567DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_55)—Offset 25Ch 1533
16.568DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_56)—Offset 260h 1533
16.569DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_57)—Offset 264h 1534
16.570DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_58)—Offset 268h 1534
16.571DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_59)—Offset 26Ch 1534
16.572DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_60)—Offset 270h 1535
16.573DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_61)—Offset 274h 1535
16.574DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_62)—Offset 278h 1535
16.575DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_63)—Offset 27Ch 1536
16.576DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_64)—Offset 280h 1536
16.577DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_65)—Offset 284h 1536
16.578DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_66)—Offset 288h 1537
16.579DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_67)—Offset 28Ch 1537
16.580DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_68)—Offset 290h 1537
16.581DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_69)—Offset 294h 1538
16.582DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_70)—Offset 298h 1538
16.583DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_71)—Offset 29Ch 1538
16.584DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_72)—Offset 2A0h 1539
16.585DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_73)—Offset 2A4h 1539
16.586DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_74)—Offset 2A8h 1539

16.587DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_75)—Offset 2ACh 1540
16.588DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_76)—Offset 2B0h 1540
16.589DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_77)—Offset 2B4h 1540
16.590DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_78)—Offset 2B8h 1541
16.591DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_79)—Offset 2BCh 1541
16.592DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_80)—Offset 2C0h 1541
16.593DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_81)—Offset 2C4h 1542
16.594DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_82)—Offset 2C8h 1542
16.595DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_83)—Offset 2CCh 1542
16.596DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_84)—Offset 2D0h 1543
16.597DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_85)—Offset 2D4h 1543
16.598DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_86)—Offset 2D8h 1543
16.599DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_87)—Offset 2DCh 1544
16.600DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_88)—Offset 2E0h 1544
16.601DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_89)—Offset 2E4h 1544
16.602DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_90)—Offset 2E8h 1545
16.603DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_91)—Offset 2ECh 1545
16.604DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_92)—Offset 2F0h.....1545
16.605DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_93)—Offset 2F4h.....1546
16.606DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_94)—Offset 2F8h.....1546
16.607DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_95)—Offset 2FCh 1546
16.608DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_96)—Offset 300h 1547
16.609DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_97)—Offset 304h 1547
16.610DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_98)—Offset 308h 1547
16.611DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_99)—Offset 30Ch 1548
16.612DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_100)—Offset 310h 1548
16.613DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_101)—Offset 314h 1549
16.614DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_102)—Offset 318h 1549
16.615DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_103)—Off-

set 31Ch 1549
16.616DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_104)—Offset 320h 1550
16.617DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_105)—Offset 324h 1550
16.618DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_106)—Offset 328h 1550
16.619DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_107)—Offset 32Ch 1551
16.620DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_108)—Offset 330h 1551
16.621DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_109)—Offset 334h 1552
16.622DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_110)—Offset 338h 1552
16.623DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_111)—Offset 33Ch 1552
16.624DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_112)—Offset 340h 1553
16.625DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_113)—Offset 344h 1553
16.626DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_114)—Offset 348h 1553
16.627DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_115)—Offset 34Ch 1554
16.628DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_116)—Offset 350h 1554
16.629DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_117)—Offset 354h 1555
16.630DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_118)—Offset 358h 1555
16.631DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_119)—Offset 35Ch 1555
16.632DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_120)—Offset 360h 1556
16.633DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_121)—Offset 364h 1556
16.634DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_122)—Offset 368h 1556
16.635DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_123)—Offset 36Ch 1557
16.636DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_124)—Offset 370h 1557
16.637DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_125)—Offset 374h 1558
16.638DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_126)—Offset 378h 1558
16.639DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_127)—Offset 37Ch 1558
16.640DEBUG_RESPONSE_HEADER_STACK_REG (DEBUG_RESPONSE_HEADER_STACK_REG_0)—Offset 380h 1559
16.641DEBUG_RESPONSE_HEADER_STACK_REG (DEBUG_RESPONSE_HEADER_STACK_REG_1)—Offset 384h 1559
16.642DEBUG_RESPONSE_HEADER_STACK_REG (DEBUG_RESPONSE_HEADER_STACK_REG_2)—Offset 388h 1559
16.643Vendor ID (VID)—Offset 0h 1560
16.644Device ID (DID)—Offset 2h 1560

16.645Command (CMD)—Offset 4h 1560
16.646Device Status (STS)—Offset 6h 1561
16.647Revision ID (RID)—Offset 8h 1562
16.648Programming Interface (PI)—Offset 9h 1563
16.649Sub Class Code (SCC)—Offset Ah 1563
16.650Base Class Code (BCC)—Offset Bh 1563
16.651Master Latency Timer (MLT)—Offset Dh 1564
16.652Header Type (HT)—Offset Eh 1564
16.653Memory Base Address (MBAR)—Offset 10h 1564
16.654USB Subsystem Vendor ID (SSVID)—Offset 2Ch 1565
16.655USB Subsystem ID (SSID)—Offset 2Eh 1565
16.656Capabilities Pointer (CAP_PTR)—Offset 34h 1566
16.657Interrupt Line (ILINE)—Offset 3Ch 1566
16.658Interrupt Pin (IPIN)—Offset 3Dh 1566
16.659XHC System Bus Configuration 1 (XHCC1)—Offset 40h 1567
16.660XHC System Bus Configuration 2 (XHCC2)—Offset 44h 1568
16.661Clock Gating (XHCLKGTEN)—Offset 50h 1570
16.662Audio Time Synchronization (AUDSYNC)—Offset 58h 1573
16.663Serial Bus Release Number (SBRN)—Offset 60h 1573
16.664Frame Length Adjustment (FLADJ)—Offset 61h 1574
16.665Best Effort Service Latency (BESL)—Offset 62h 1574
16.666PCI Power Management Capability ID (PM_CID)—Offset 70h 1575
16.667Next Item Pointer #1 (PM_NEXT)—Offset 71h 1575
16.668Power Management Capabilities (PM_CAP)—Offset 72h 1576
16.669Power Management Control/Status (PM_CS)—Offset 74h 1576
16.670Message Signaled Interrupt CID (MSI_CID)—Offset 80h 1577
16.671Next item pointer (MSI_NEXT)—Offset 81h 1578
16.672Message Signaled Interrupt Message Control (MSI_MCTL)—Offset 82h 1578
16.673Message Signaled Interrupt Message Address (MSI_MAD)—Offset 84h 1579
16.674Message Signaled Interrupt Upper Address (MSI_MUAD)—Offset 88h 1579
16.675Message Signaled Interrupt Message Data (MSI_MD)—Offset 8Ch 1579
16.676Device Idle Capability (DEVIDLE)—Offset 90h 1580
16.677Vendor Specific Header (VSHDR)—Offset 94h 1580
16.678SW LTR POINTER (SWLTRPTR)—Offset 98h 1581
16.679Device Idle Pointer Register (DEVIDLEPTR)—Offset 9Ch 1581
16.680Device Idle Power ON Latency (DEVIDLEPOL)—Offset A0h 1582
16.681High Speed Configuration 2 (HSCFG2)—Offset A4h 1583
16.682XHCI USB2 Overcurrent Pin Mapping 1 (U2OCM1)—Offset B0h 1584
16.683XHCI USB2 Overcurrent Pin Mapping 2 (U2OCM2)—Offset B4h 1584
16.684XHCI USB3 Overcurrent Pin Mapping 1 (U3OCM1)—Offset D0h 1584
16.685XHCI USB3 Overcurrent Pin Mapping 2 (U3OCM2)—Offset D4h 1585
16.686XHCC3 (XHCC3)—Offset FCh 1585
16.687Capability Registers Length (CAPLENGTH)—Offset 0h 1586
16.688Host Controller Interface Version Number (HCIVERSION)—Offset 2h 1586
16.689Structural Parameters 1 (HCSPARAMS1)—Offset 4h 1586
16.690Structural Parameters 2 (HCSPARAMS2)—Offset 8h 1587
16.691Structural Parameters 3 (HCSPARAMS3)—Offset Ch 1587
16.692Capability Parameters (HCCPARAMS)—Offset 10h 1588
16.693Doorbell Offset (DBOFF)—Offset 14h 1589
16.694Runtime Register Space Offset (RTSOFF)—Offset 18h 1589
16.695USB Command (USBCMD)—Offset 80h 1589
16.696USB Status (USBSTS)—Offset 84h 1590
16.697Page Size (PAGESIZE)—Offset 88h 1591
16.698Device Notification Control (DNCTRL)—Offset 94h 1591
16.699Command Ring Low (CRCR_LO)—Offset 98h 1591

16.700Command Ring High (CRCR_HI)—Offset 9Ch 1592
16.701Device Context Base Address Array Pointer Low (DCBAAP_LO)—Offset B0h..... 1592
16.702Device Context Base Address Array Pointer High (DCBAAP_HI)—Offset B4h 1592
16.703Configure (CONFIG)—Offset B8h 1593
16.704Port Status and Control USB2 (PORTSC1)—Offset 480h 1593
16.705Port Power Management Status and Control USB2 (PORTPMSC1)—Offset 484h..... 1594
16.706Port X Hardware LPM Control Register (PORTHLPMC1)—Offset 48Ch..... 1595
16.707Port Status and Control USB2 (PORTSC2)—Offset 490h 1596
16.708Port Power Management Status and Control USB2 (PORTPMSC2)—Offset 494h..... 1597
16.709Port X Hardware LPM Control Register (PORTHLPMC2)—Offset 49Ch..... 1598
16.710Port Status and Control USB2 (PORTSC3)—Offset 4A0h 1598
16.711Port Power Management Status and Control USB2 (PORTPMSC3)—Offset 4A4h..... 1600
16.712Port X Hardware LPM Control Register (PORTHLPMC3)—Offset 4ACh 1600
16.713Port Status and Control USB2 (PORTSC4)—Offset 4B0h 1601
16.714Port Power Management Status and Control USB2 (PORTPMSC4)—Offset 4B4h..... 1602
16.715Port X Hardware LPM Control Register (PORTHLPMC4)—Offset 4BCh 1603
16.716Port Status and Control USB2 (PORTSC5)—Offset 4C0h 1603
16.717Port Power Management Status and Control USB2 (PORTPMSC5)—Offset 4C4h..... 1605
16.718Port X Hardware LPM Control Register (PORTHLPMC5)—Offset 4CCh 1605
16.719Port Status and Control USB2 (PORTSC6)—Offset 4D0h..... 1606
16.720Port Power Management Status and Control USB2 (PORTPMSC6)—Offset 4D4h 1607
16.721Port X Hardware LPM Control Register (PORTHLPMC6)—Offset 4DCh 1608
16.722Port Status and Control USB2 (PORTSC7)—Offset 4E0h 1609
16.723Port Power Management Status and Control USB2 (PORTPMSC7)—Offset 4E4h..... 1610
16.724Port X Hardware LPM Control Register (PORTHLPMC7)—Offset 4ECh..... 1611
16.725Port Status and Control USB2 (PORTSC8)—Offset 4F0h 1611
16.726Port Power Management Status and Control USB2 (PORTPMSC8)—Offset 4F4h..... 1613
16.727Port X Hardware LPM Control Register (PORTHLPMC8)—Offset 4FCh..... 1613
16.728Port Status and Control USB2 (PORTSC9)—Offset 500h 1614
16.729Port Power Management Status and Control USB2 (PORTPMSC9)—Offset 504h..... 1615
16.730Port X Hardware LPM Control Register (PORTHLPMC9)—Offset 50Ch..... 1616
16.731Port Status and Control USB3 (PORTSC10)—Offset 510h 1616
16.732Port Power Management Status and Control USB3 (PORTPMSC10)—Offset 514h 1618
16.733USB3 Port Link Info (PORTLI10)—Offset 518h 1618
16.734Port Status and Control USB3 (PORTSC11)—Offset 520h 1618
16.735Port Power Management Status and Control USB3 (PORTPMSC11)—Offset 524h 1620
16.736USB3 Port Link Info (PORTLI11)—Offset 528h 1620
16.737Port Status and Control USB3 (PORTSC12)—Offset 530h 1620
16.738Port Power Management Status and Control USB3 (PORTPMSC12)—Offset 534h 1622
16.739USB3 Port Link Info (PORTLI12)—Offset 538h 1622
16.740Port Status and Control USB3 (PORTSC13)—Offset 540h 1622
16.741Port Power Management Status and Control USB3 (PORTPMSC13)—Offset 544h 1624
16.742USB3 Port Link Info (PORTLI13)—Offset 548h 1624
16.743Port Status and Control USB3 (PORTSC14)—Offset 550h 1624
16.744Port Power Management Status and Control USB3 (PORTPMSC14)—Offset 554h 1626
16.745USB3 Port Link Info (PORTLI14)—Offset 558h 1626
16.746Port Status and Control USB3 (PORTSC15)—Offset 560h 1626
16.747Port Power Management Status and Control USB3 (PORTPMSC15)—Offset 564h 1628
16.748USB3 Port Link Info (PORTLI15)—Offset 568h 1628
16.749Port Status and Control USB3 (PORTSC16)—Offset 570h 1628
16.750Port Power Management Status and Control USB3 (PORTPMSC16)—Offset 574h 1630
16.751USB3 Port Link Info (PORTLI16)—Offset 578h 1630
16.752Microframe Index (RTMFINDEX)—Offset 2000h 1630
16.753Interrupter 1 Management (IMAN0)—Offset 2020h 1631
16.754Interrupter 1 Moderation (IMOD0)—Offset 2024h 1631

16.755Event Ring Segment Table Size 1 (ERSTSZ0)—Offset 2028h 1632
16.756Event Ring Segment Table Base Address Low 1 (ERSTBA_LO0)—Offset 2030h 1632
16.757Event Ring Segment Table Base Address High 1 (ERSTBA_HI0)—Offset 2034h 1632
16.758Event Ring Dequeue Pointer Low 1 (ERDP_LO0)—Offset 2038h 1633
16.759Event Ring Dequeue Pointer High 1 (ERDP_HI0)—Offset 203Ch 1633
16.760Interrupter 2 Management (IMAN1)—Offset 2040h 1633
16.761Interrupter 2 Moderation (IMOD1)—Offset 2044h 1634
16.762Event Ring Segment Table Size 2 (ERSTSZ1)—Offset 2048h 1634
16.763Event Ring Segment Table Base Address Low 2 (ERSTBA_LO1)—Offset 2050h 1634
16.764Event Ring Segment Table Base Address High 2 (ERSTBA_HI1)—Offset 2054h 1635
16.765Event Ring Dequeue Pointer Low 2 (ERDP_LO1)—Offset 2058h 1635
16.766Event Ring Dequeue Pointer High 2 (ERDP_HI1)—Offset 205Ch 1636
16.767Interrupter 3 Management (IMAN2)—Offset 2060h 1636
16.768Interrupter 3 Moderation (IMOD2)—Offset 2064h 1636
16.769Event Ring Segment Table Size 3 (ERSTSZ2)—Offset 2068h 1637
16.770Event Ring Segment Table Base Address Low 3 (ERSTBA_LO2)—Offset 2070h 1637
16.771Event Ring Segment Table Base Address High 3 (ERSTBA_HI2)—Offset 2074h 1637
16.772Event Ring Dequeue Pointer Low 3 (ERDP_LO2)—Offset 2078h 1638
16.773Event Ring Dequeue Pointer High 3 (ERDP_HI2)—Offset 207Ch 1638
16.774Interrupter 4 Management (IMAN3)—Offset 2080h 1638
16.775Interrupter 4 Moderation (IMOD3)—Offset 2084h 1639
16.776Event Ring Segment Table Size 4 (ERSTSZ3)—Offset 2088h 1639
16.777Event Ring Segment Table Base Address Low 4 (ERSTBA_LO3)—Offset 2090h 1639
16.778Event Ring Segment Table Base Address High 4 (ERSTBA_HI3)—Offset 2094h 1640
16.779Event Ring Dequeue Pointer Low 4 (ERDP_LO3)—Offset 2098h 1640
16.780Event Ring Dequeue Pointer High 4 (ERDP_HI3)—Offset 209Ch 1641
16.781Interrupter 5 Management (IMAN4)—Offset 20A0h 1641
16.782Interrupter 5 Moderation (IMOD4)—Offset 20A4h 1641
16.783Event Ring Segment Table Size 5 (ERSTSZ4)—Offset 20A8h 1642
16.784Event Ring Segment Table Base Address Low 5 (ERSTBA_LO4)—Offset 20B0h 1642
16.785Event Ring Segment Table Base Address High 5 (ERSTBA_HI4)—Offset 20B4h 1642
16.786Event Ring Dequeue Pointer Low 5 (ERDP_LO4)—Offset 20B8h 1643
16.787Event Ring Dequeue Pointer High 5 (ERDP_HI4)—Offset 20BCh 1643
16.788Interrupter 6 Management (IMAN5)—Offset 20C0h 1643
16.789Interrupter 6 Moderation (IMOD5)—Offset 20C4h 1644
16.790Event Ring Segment Table Size 6 (ERSTSZ5)—Offset 20C8h 1644
16.791Event Ring Segment Table Base Address Low 6 (ERSTBA_LO5)—Offset 20D0h 1645
16.792Event Ring Segment Table Base Address High 6 (ERSTBA_HI5)—Offset 20D4h 1645
16.793Event Ring Dequeue Pointer Low 6 (ERDP_LO5)—Offset 20D8h 1645
16.794Event Ring Dequeue Pointer High 6 (ERDP_HI5)—Offset 20DCh 1646
16.795Interrupter 7 Management (IMAN6)—Offset 20E0h 1646
16.796Interrupter 7 Moderation (IMOD6)—Offset 20E4h 1646
16.797Event Ring Segment Table Size 7 (ERSTSZ6)—Offset 20E8h 1647
16.798Event Ring Segment Table Base Address Low 7 (ERSTBA_LO6)—Offset 20F0h 1647
16.799Event Ring Segment Table Base Address High 7 (ERSTBA_HI6)—Offset 20F4h 1647
16.800Event Ring Dequeue Pointer Low 7 (ERDP_LO6)—Offset 20F8h 1648
16.801Event Ring Dequeue Pointer High 7 (ERDP_HI6)—Offset 20FCh 1648
16.802Interrupter 8 Management (IMAN7)—Offset 2100h 1648
16.803Interrupter 8 Moderation (IMOD7)—Offset 2104h 1649
16.804Event Ring Segment Table Size 8 (ERSTSZ7)—Offset 2108h 1649
16.805Event Ring Segment Table Base Address Low 8 (ERSTBA_LO7)—Offset 2110h 1650
16.806Event Ring Segment Table Base Address High 8 (ERSTBA_HI7)—Offset 2114h 1650
16.807Event Ring Dequeue Pointer Low 8 (ERDP_LO7)—Offset 2118h 1650
16.808Event Ring Dequeue Pointer High 8 (ERDP_HI7)—Offset 211Ch 1651
16.809Door Bell 1 (DB0)—Offset 3000h 1651

16.810Door Bell 2 (DB1)—Offset 3004h 1651
16.811Door Bell 3 (DB2)—Offset 3008h 1652
16.812Door Bell 4 (DB3)—Offset 300Ch 1652
16.813Door Bell 5 (DB4)—Offset 3010h 1653
16.814Door Bell 6 (DB5)—Offset 3014h 1653
16.815Door Bell 7 (DB6)—Offset 3018h 1654
16.816Door Bell 8 (DB7)—Offset 301Ch 1654
16.817Door Bell 9 (DB8)—Offset 3020h 1654
16.818Door Bell 10 (DB9)—Offset 3024h 1655
16.819Door Bell 11 (DB10)—Offset 3028h 1655
16.820Door Bell 12 (DB11)—Offset 302Ch 1656
16.821Door Bell 13 (DB12)—Offset 3030h 1656
16.822Door Bell 14 (DB13)—Offset 3034h 1656
16.823Door Bell 15 (DB14)—Offset 3038h 1657
16.824Door Bell 16 (DB15)—Offset 303Ch 1657
16.825Door Bell 17 (DB16)—Offset 3040h 1658
16.826Door Bell 18 (DB17)—Offset 3044h 1658
16.827Door Bell 19 (DB18)—Offset 3048h 1658
16.828Door Bell 20 (DB19)—Offset 304Ch 1659
16.829Door Bell 21 (DB20)—Offset 3050h 1659
16.830Door Bell 22 (DB21)—Offset 3054h 1660
16.831Door Bell 23 (DB22)—Offset 3058h 1660
16.832Door Bell 24 (DB23)—Offset 305Ch 1660
16.833Door Bell 25 (DB24)—Offset 3060h 1661
16.834Door Bell 26 (DB25)—Offset 3064h 1661
16.835Door Bell 27 (DB26)—Offset 3068h 1662
16.836Door Bell 28 (DB27)—Offset 306Ch 1662
16.837Door Bell 29 (DB28)—Offset 3070h 1662
16.838Door Bell 30 (DB29)—Offset 3074h 1663
16.839Door Bell 31 (DB30)—Offset 3078h 1663
16.840Door Bell 32 (DB31)—Offset 307Ch 1664
16.841Door Bell 32 (DB32)—Offset 3080h 1664
16.842XECP_SUPP_USB2_0 (XECP_SUPP_USB2_0)—Offset 8000h 1664
16.843XECP_SUPP_USB2_1 (XECP_SUPP_USB2_1)—Offset 8004h 1665
16.844XECP_SUPP_USB2_2 (XECP_SUPP_USB2_2)—Offset 8008h 1665
16.845XECP_SUPP_USB2_3 (Full Speed) (XECP_SUPP_USB2_3)—Offset 8010h 1666
16.846XECP_SUPP_USB2_4 (Low Speed) (XECP_SUPP_USB2_4)—Offset 8014h 1666
16.847XECP_SUPP_USB2_5 (High Speed) (XECP_SUPP_USB2_5)—Offset 8018h 1667
16.848XECP_SUPP_USB3_0 (XECP_SUPP_USB3_0)—Offset 8020h 1667
16.849XECP_SUPP_USB3_1 (XECP_SUPP_USB3_1)—Offset 8024h 1668
16.850XECP_SUPP_USB3_2 (XECP_SUPP_USB3_2)—Offset 8028h 1668
16.851XECP_SUPP_USB3_3 (XECP_SUPP_USB3_3)—Offset 8030h 1669
16.852XECP_SUPP_USB3_4 (XECP_SUPP_USB3_4)—Offset 8034h 1669
16.853XECP_SUPP_USB3_5 (XECP_SUPP_USB3_5)—Offset 8038h 1670
16.854XECP_SUPP_USB3_6 (XECP_SUPP_USB3_6)—Offset 803Ch 1670
16.855XECP_SUPP_USB3_7 (XECP_SUPP_USB3_7)—Offset 8040h 1671
16.856XECP_SUPP_USB3_8 (XECP_SUPP_USB3_8)—Offset 8044h 1671
16.857XECP_SUPP_USB3_9 (XECP_SUPP_USB3_9)—Offset 8048h 1672
16.858Host Controller Capability (HOST_CTRL_CAP_REG)—Offset 8070h 1672
16.859Override EP Flow Control (HOST_CLR_MASK_REG)—Offset 8078h 1673
16.860Clear Active IN EP ID Control (HOST_CLR_IN_EP_VALID_REG)—Offset 807Ch 1673
16.861Clear Poll Mask Control (HOST_CLR_PMASK_REG)—Offset 8080h 1673
16.862Host Control Scheduler (HOST_CTRL_SCH_REG)—Offset 8094h 1674
16.863Global Port Control (HOST_CTRL_PORT_CTRL)—Offset 80A0h 1675
16.864Power Management Control (PMCTRL_REG)—Offset 80A4h 1675

16.865PGCB Control (PGCBCTRL_REG)—Offset 80A8h 1677
16.866D0I3 Control (DOI3CTRL_REG)—Offset 80ACh 1680
16.867HOST_CTRL_MISC_REG (HOST_CTRL_MISC_REG)—Offset 80B0h 1681
16.868HOST_CTRL_MISC_REG2 (HOST_CTRL_MISC_REG2)—Offset 80B4h 1682
16.869SSPE_REG (SSPE_REG)—Offset 80B8h 1683
16.870(SSPITPE)—Offset 80BCh 1683
16.871AUX Reset Control (AUX_CTRL_REG)—Offset 80C0h 1684
16.872Super Speed Bandwidth Overload (HOST_BW_OV_SS_REG)—Offset 80C4h 1686
16.873High Speed TT Bandwidth Overload (HOST_BW_OV_HS_REG)—Offset 80C8h 1686
16.874Bandwidth Overload Full Low Speed (HOST_BW_OV_FS_LS_REG)—Offset 80CCh ... 1687
16.875System Bandwidth Overload (HOST_BW_OV_SYS_REG)—Offset 80D0h 1687
16.876Scheduler Async Delay (HOST_CTRL_SCH_ASYNC_DELAY_REG)—Offset 80D4h 1688
16.877DEVICE MODE CONTROL REG 0 (DUAL_ROLE_CFG_REG0)—Offset 80D8h 1688
16.878DEVICE MODE CONTROL REG 1 (DUAL_ROLE_CFG_REG1)—Offset 80DCh 1690
16.879AUX Power Management Control (AUX_CTRL_REG1)—Offset 80E0h 1691
16.880Battery Charge (BATTERY_CHARGE_REG)—Offset 80E4h 1693
16.881Port Watermark (HOST_CTRL_WATERMARK_REG)—Offset 80E8h 1693
16.882SuperSpeed Port Link Control (HOST_CTRL_PORT_LINK_REG)—Offset 80ECh 1694
16.883USB2 Port Link Control 1 (USB2_LINK_MGR_CTRL_REG1)—Offset 80F0h 1695
16.884USB2 Port Link Control 2 (USB2_LINK_MGR_CTRL_REG2)—Offset 80F4h 1697
16.885USB2 Port Link Control 3 (USB2_LINK_MGR_CTRL_REG3)—Offset 80F8h 1697
16.886USB2 Port Link Control 4 (USB2_LINK_MGR_CTRL_REG4)—Offset 80FCh 1698
16.887Bandwidth Calc Control (HOST_CTRL_BW_CTRL_REG)—Offset 8100h 1698
16.888Host Interface Control (HOST_IF_CTRL_REG)—Offset 8108h 1698
16.889Bandwidth Overload Burst (HOST_BW_OV_BURST_REG)—Offset 810Ch 1699
16.890USB Max Bandwidth Control 4 (HOST_CTRL_BW_MAX_REG)—Offset 8128h 1699
16.891USB2 Linestate Debug (LINESTATE_DEBUG_REG)—Offset 8130h 1700
16.892USB2 Protocol Gap Timer (USB2_PROTOCOL_GAP_TIMER_REG)—Offset 8134h 1701
16.893USB2 Protocol Bus Timeout Timer (USB2_PROTOCOL_BTO_TIMER_REG)—Offset 813Ch. 1701
16.894Power Scheduler Control-0 (PWR_SCHED_CTRL0)—Offset 8140h 1702
16.895Power Scheduler Control-2 (PWR_SCHED_CTRL2)—Offset 8144h 1702
16.896AUX Power Management Control (AUX_CTRL_REG2)—Offset 8154h 1703
16.897USB2 PHY Power Management Control (USB2_PHY_PMC)—Offset 8164h 1705
16.898USB Power Gating Control (USB_PGC)—Offset 8168h 1706
16.899xHCI Aux Clock Control Register (XHCI_AUX_CCR)—Offset 816Ch 1707
16.900USB LPM Parameters (USB_LPM_PARAM)—Offset 8170h 1709
16.901xHC Latency Tolerance Parameters - LTV Control (XLTP_LTV1)—Offset 8174h 1710
16.902xHC Latency Tolerance Parameters LTV Control 2 (XLTP_LTV2)—Offset 8178h 1710
16.903xHC Latency Tolerance Parameters - High Idle Time Control (XLTP_HITC)—Offset 817Ch 1711
16.904xHC Latency Tolerance Parameters - Medium Idle Time Control (XLTP_MITC)—Offset 8180h 1712
16.905xHC Latency Tolerance Parameters Low Idle Time Control (XLTP_LITC)—Offset 8184h ... 1712
16.906HOST_CTRL_BW_MAX3_REG (HOST_CTRL_BW_MAX3_REG)—Offset 8188h 1713
16.907THRM_HOST_CTRL_REG (THRM_HOST_CTRL_REG)—Offset 819Ch 1713
16.908LFPS_PM_CTRL_REG (LFPS_PM_CTRL_REG)—Offset 81A0h 1715
16.909U2PDM (U2PDM)—Offset 81A4h 1715
16.910U2PCM (U2PCM)—Offset 81A8h 1715
16.911U3PDM (U3PDM)—Offset 81ACh 1716
16.912U3PCM (U3PCM)—Offset 81B0h 1716
16.913THRM_HOST_CTRL_REG2 (THRM_HOST_CTRL_REG2)—Offset 81B4h 1716
16.914LFPSONCOUNT_REG (LFPSONCOUNT_REG)—Offset 81B8h 1717
16.915(USB2PMCTRL_REG)—Offset 81C4h 1717

16.916ECC_PARITY_ERROR_LOG_REG (ECC_PARITY_ERROR_LOG_REG)—Offset 83F8h ... 1718
16.917ECC_POISONING_CTRL_REG (ECC_POISONING_CTRL_REG)—Offset 83FCh ... 1720
16.918USB2_PORT_STATE_REG (USB2_PORT_STATE_REG)—Offset 8400h ... 1721
16.919USB3_PORT_STATE_REG (USB3_PORT_STATE_REG)—Offset 8408h ... 1721
16.920FUS1_REG (FUS1_REG)—Offset 8410h ... 1722
16.921FUS2_REG (FUS2_REG)—Offset 8414h ... 1723
16.922FUS3_REG (FUS3_REG)—Offset 8418h ... 1723
16.923STRAP1_REG (STRAP1_REG)—Offset 841Ch ... 1723
16.924STRAP2_REG (STRAP2_REG)—Offset 8420h ... 1724
16.925STRAP3_REG (STRAP3_REG)—Offset 8424h ... 1724
16.926DFT_REG1 (DFT_REG1)—Offset 8430h ... 1725
16.927DFT_REG2 (DFT_REG2)—Offset 8434h ... 1725
16.928DFT_REG3 (DFT_REG3)—Offset 8438h ... 1726
16.929dft_reg4 (DFT_REG4)—Offset 843Ch ... 1727
16.930dft_reg5 (DFT_REG5)—Offset 8440h ... 1727
16.931XECP_CMDM_STS0 (XECP_CMDM_STS0)—Offset 8448h ... 1728
16.932XECP_CMDM_STS1 (XECP_CMDM_STS1)—Offset 844Ch ... 1729
16.933XECP_CMDM_STS2 (XECP_CMDM_STS2)—Offset 8450h ... 1730
16.934XECP_CMDM_STS3 (XECP_CMDM_STS3)—Offset 8454h ... 1730
16.935XECP_CMDM_STS4 (XECP_CMDM_STS4)—Offset 8458h ... 1730
16.936XECP_CMDM_STS5 (XECP_CMDM_STS5)—Offset 845Ch ... 1730
16.937AUX Power PHY Reset (UPORTS_PON_RST_REG)—Offset 8460h ... 1731
16.938Latency Tolerance Control 0 (HOST_IF_LAT_TOL_CTRL_REG0)—Offset 8464h ... 1731
16.939USB Legacy Support Capability (USBLEGSUP)—Offset 846Ch ... 1732
16.940USB Legacy Support Control Status (USBLEGCTLSTS)—Offset 8470h ... 1733
16.941Port Disable Override capability register (PDO_CAPABILITY)—Offset 84F4h ... 1733
16.942USB2 Port Disable Override (USB2PDO)—Offset 84F8h ... 1734
16.943USB3 Port Disable Override (USB3PDO)—Offset 84FCh ... 1734
16.944HW state capability register (HW_STATE_CAPABILITY)—Offset 8500h ... 1734
16.945HW state register 1 (HW_STATE_REG1)—Offset 8504h ... 1735
16.946HW state register 2 (HW_STATE_REG2)—Offset 8508h ... 1735
16.947HW state register 3 (HW_STATE_REG3)—Offset 850Ch ... 1735
16.948HW state register 4 (HW_STATE_REG4)—Offset 8510h ... 1736
16.949CONFIG mirror capability register (CONFIG_MIRROR_CAPABILITY)—Offset 8600h ... 1736
16.950Command (CMD_MMIO)—Offset 8604h ... 1736
16.951Device Status (STS_MMIO)—Offset 8606h ... 1737
16.952Revision ID (RID_MMIO)—Offset 8608h ... 1738
16.953Programming Interface (PI_MMIO)—Offset 8609h ... 1739
16.954Sub Class Code (SCC_MMIO)—Offset 860Ah ... 1739
16.955Base Class Code (BCC_MMIO)—Offset 860Bh ... 1739
16.956Master Latency Timer (MLT_MMIO)—Offset 860Dh ... 1740
16.957Header Type (HT_MMIO)—Offset 860Eh ... 1740
16.958Memory Base Address (MBAR_MMIO)—Offset 8610h ... 1740
16.959USB Subsystem Vendor ID (SSVID_MMIO)—Offset 862Ch ... 1741
16.960USB Subsystem ID (SSID_MMIO)—Offset 862Eh ... 1741
16.961Capabilities Pointer (CAP_PTR_MMIO)—Offset 8634h ... 1742
16.962Interrupt Line (ILINE_MMIO)—Offset 863Ch ... 1742
16.963Interrupt Pin (IPIN_MMIO)—Offset 863Dh ... 1742
16.964XHC System Bus Configuration 1 (XHCC1_MMIO)—Offset 8640h ... 1743
16.965XHC System Bus Configuration 2 (XHCC2_MMIO)—Offset 8644h ... 1744
16.966Clock Gating (XHCLKGTEN_MMIO)—Offset 8650h ... 1746
16.967Audio Time Synchronization (AUDSYNC_MMIO)—Offset 8658h ... 1749
16.968Serial Bus Release Number (SBRN_MMIO)—Offset 8660h ... 1750
16.969Frame Length Adjustment (FLADJ_MMIO)—Offset 8661h ... 1750
16.970Best Effort Service Latency (BESL_MMIO)—Offset 8662h ... 1751

16.971PCI Power Management Capability ID (PM_CID_MMIO)—Offset 8670h 1751
16.972Next Item Pointer #1 (PM_NEXT_MMIO)—Offset 8671h 1751
16.973Power Management Capabilities (PM_CAP_MMIO)—Offset 8672h.....1752
16.974Power Management Control/Status (PM_CS_MMIO)—Offset 8674h 1753
16.975Message Signaled Interrupt CID (MSI_CID_MMIO)—Offset 8680h 1754
16.976Next item pointer (MSI_NEXT_MMIO)—Offset 8681h 1754
16.977Message Signaled Interrupt Message Control (MSI_MCTL_MMIO)—Offset 8682h.....1754
16.978Message Signaled Interrupt Message Address (MSI_MAD_MMIO)—Offset 8684h.....1755
16.979Message Signaled Interrupt Upper Address (MSI_MUAD_MMIO)—Offset 8688h 1755
16.980Message Signaled Interrupt Message Data (MSI_MD_MMIO)—Offset 868Ch.....1756
16.981Device Idle Capability (DEVIDLE_MMIO)—Offset 8690h.....1756
16.982Vendor Specific Header (VSHDR_MMIO)—Offset 8694h.....1757
16.983SW LTR POINTER (SWLTRPTR_MMIO)—Offset 8698h.....1757
16.984Device Idle Pointer Register (DEVIDLEPTR_MMIO)—Offset 869Ch.....1758
16.985Device Idle Power ON Latency (DEVIDLEPOL_MMIO)—Offset 86A0h.....1759
16.986High Speed Configuration 2 (HSCFG2_MMIO)—Offset 86A4h 1759
16.987XHCI USB2 Overcurrent Pin Mapping 1 (U2OCM1_MMIO)—Offset 86B0h 1761
16.988XHCI USB2 Overcurrent Pin Mapping 2 (U2OCM2_MMIO)—Offset 86B4h 1761
16.989XHCI USB3 Overcurrent Pin Mapping 1 (U3OCM1_MMIO)—Offset 86D0h 1761
16.990XHCI USB3 Overcurrent Pin Mapping 2 (U3OCM2_MMIO)—Offset 86D4h 1762
16.991XHCC3 (XHCC3_MMIO)—Offset 86FCh.....1762
16.992Debug Capability ID Register (DCID)—Offset 8700h 1763
16.993Debug Capability Doorbell Register (DCDB)—Offset 8704h 1763
16.994Debug Capability Event Ring Segment Table Size Register (DCERSTSZ)—Offset 8708h... 1764
16.995Debug Capability Event Ring Segment Table Base Address Register (DCERSTBA)—Offset 8710h.....1764
16.996Debug Capability Event Ring Dequeue Pointer Register (DCERDP)—Offset 8718h....1765
16.997Debug Capability Control Register (DCCTRL)—Offset 8720h.....1765
16.998Debug Capability Status Register (DCST)—Offset 8724h 1766
16.999Debug Capability Port Status and Control Register (DCPORTSC)—Offset 8728h 1766
16.1000Debug Capability Context Pointer Register (DCCP)—Offset 8730h 1768
16.1001Debug Capability Device Descriptor Info Register 1 (DCDDI1)—Offset 8738h 1769
16.1002Debug Capability Device Descriptor Info Register 2 (DCDDI2)—Offset 873Ch 1769
16.1003Debug Capability Descriptor Parameters (DCDP)—Offset 8740h.....1769
16.1004Debug Device Control ODMA (DBGDEV_CTRL_ODMA_REG)—Offset 8748h 1770
16.1005DBC Control Register 1 (DBCCTL_REG)—Offset 8760h 1771
16.1006SSIC Policy and Implementation Specific Registers Capability ID register (SSIC_IMPLE-MENTATION_CAPABILITY_ID_REG)—Offset 8800h.....1772
16.1007SSIC global configuration control register (SSIC_GLOBAL_CONFIG_CONTROL_REG)—Offset 8804h 1772
16.1008 (PORT1_SSIC_CONFIG_REG1)—Offset 8808h 1773
16.1009 (PORT1_SSIC_CONFIG_REG2)—Offset 880Ch.....1774
16.1010 (PORT1_SSIC_CONFIG_REG3)—Offset 8810h 1775
16.1011 (PORT1_SSIC_CONFIG_REG4)—Offset 8814h 1776
16.1012 (PORT1_SSIC_LOOPBACK_CONFIG_REG)—Offset 8818h 1776
16.1013 (PORT1_SSIC_LOOPBACK_BURST_COUNT_REG)—Offset 881Ch.....1776
16.1014 (PORT1_SSIC_LOOPBACK_ERROR_COUNT_REG)—Offset 8820h.....1777
16.1015SSIC Local and Remote Profile Registers Capability ID register (SSIC_PROFILE_CAPA-BILITY_ID_REG)—Offset 8900h.....1778
16.1016SSIC Port N Register Access Control (PORT1_REGISTER_ACCESS_CONTROL)—Offset 8904h.....1778
16.1017SSIC Port N Register Access Status (PORT1_REGISTER_ACCESS_STATUS)—Offset 8908h.....1779
16.1018 (PORT1_PROFILE_ATTRIBUTES_REG0)—Offset 890Ch 1780

16.1019(PORT1_PROFILE_ATTRIBUTES_REG1)—Offset 8910h 1780
16.1020(PORT1_PROFILE_ATTRIBUTES_REG2)—Offset 8914h 1781
16.1021(PORT1_PROFILE_ATTRIBUTES_REG3)—Offset 8918h 1781
16.1022(PORT1_PROFILE_ATTRIBUTES_REG4)—Offset 891Ch 1782
16.1023(PORT1_PROFILE_ATTRIBUTES_REG5)—Offset 8920h 1782
16.1024(PORT1_PROFILE_ATTRIBUTES_REG6)—Offset 8924h 1783
16.1025(PORT1_PROFILE_ATTRIBUTES_REG7)—Offset 8928h 1783
16.1026(PORT1_PROFILE_ATTRIBUTES_REG8)—Offset 892Ch 1784
16.1027(PORT1_PROFILE_ATTRIBUTES_REG9)—Offset 8930h 1784
16.1028(PORT1_PROFILE_ATTRIBUTES_REG10)—Offset 8934h 1785
16.1029(PORT1_PROFILE_ATTRIBUTES_REG11)—Offset 8938h 1785
16.1030(PORT1_PROFILE_ATTRIBUTES_REG12)—Offset 893Ch 1786
16.1031(PORT1_PROFILE_ATTRIBUTES_REG13)—Offset 8940h 1786
16.1032(PORT1_PROFILE_ATTRIBUTES_REG14)—Offset 8944h 1787
16.1033(PORT1_PROFILE_ATTRIBUTES_REG15)—Offset 8948h 1787
16.1034(PORT1_PROFILE_ATTRIBUTES_REG16)—Offset 894Ch 1788
16.1035(PORT1_PROFILE_ATTRIBUTES_REG17)—Offset 8950h 1788
16.1036(PORT1_PROFILE_ATTRIBUTES_REG18)—Offset 8954h 1789
16.1037(PORT1_PROFILE_ATTRIBUTES_REG19)—Offset 8958h 1789
16.1038(PORT1_PROFILE_ATTRIBUTES_REG20)—Offset 895Ch 1790
16.1039(PORT1_PROFILE_ATTRIBUTES_REG21)—Offset 8960h 1790
16.1040(PORT1_PROFILE_ATTRIBUTES_REG22)—Offset 8964h 1791
16.1041(PORT1_PROFILE_ATTRIBUTES_REG23)—Offset 8968h 1791
16.1042(PORT1_PROFILE_ATTRIBUTES_REG24)—Offset 896Ch 1792
16.1043(PORT1_PROFILE_ATTRIBUTES_REG25)—Offset 8970h 1792
16.1044(PORT1_PROFILE_ATTRIBUTES_REG26)—Offset 8974h 1793
16.1045(PORT1_PROFILE_ATTRIBUTES_REG27)—Offset 8978h 1793
16.1046(PORT1_PROFILE_ATTRIBUTES_REG28)—Offset 897Ch 1794
16.1047(PORT1_PROFILE_ATTRIBUTES_REG29)—Offset 8980h 1794
16.1048(PORT1_PROFILE_ATTRIBUTES_REG30)—Offset 8984h 1795
16.1049(PORT1_PROFILE_ATTRIBUTES_REG31)—Offset 8988h 1795
16.1050(PORT1_PROFILE_ATTRIBUTES_REG32)—Offset 898Ch 1796
16.1051(PORT1_PROFILE_ATTRIBUTES_REG33)—Offset 8990h 1796
16.1052(PORT1_PROFILE_ATTRIBUTES_REG34)—Offset 8994h 1797
16.1053(PORT1_PROFILE_ATTRIBUTES_REG35)—Offset 8998h 1797
16.1054(PORT1_PROFILE_ATTRIBUTES_REG36)—Offset 899Ch 1798
16.1055(PORT1_PROFILE_ATTRIBUTES_REG37)—Offset 89A0h 1798
16.1056(PORT1_PROFILE_ATTRIBUTES_REG38)—Offset 89A4h 1799
16.1057(PORT1_PROFILE_ATTRIBUTES_REG39)—Offset 89A8h 1799
16.1058(PORT1_PROFILE_ATTRIBUTES_REG40)—Offset 89ACh 1800
16.1059(PORT1_PROFILE_ATTRIBUTES_REG41)—Offset 89B0h 1800
16.1060(PORT1_PROFILE_ATTRIBUTES_REG42)—Offset 89B4h 1801
16.1061(PORT1_PROFILE_ATTRIBUTES_REG43)—Offset 89B8h 1801
16.1062(PORT1_PROFILE_ATTRIBUTES_REG44)—Offset 89BCh 1802
16.1063(PORT1_PROFILE_ATTRIBUTES_REG45)—Offset 89C0h 1802
16.1064(PORT1_PROFILE_ATTRIBUTES_REG46)—Offset 89C4h 1803
16.1065(PORT1_PROFILE_ATTRIBUTES_REG47)—Offset 89C8h 1803
16.1066(PORT1_PROFILE_ATTRIBUTES_REG48)—Offset 89CCh 1804
16.1067(PORT1_PROFILE_ATTRIBUTES_REG49)—Offset 89D0h 1804
16.1068(PORT1_PROFILE_ATTRIBUTES_REG50)—Offset 89D4h 1805
16.1069(PORT1_PROFILE_ATTRIBUTES_REG51)—Offset 89D8h 1805
16.1070(PORT1_PROFILE_ATTRIBUTES_REG52)—Offset 89DCh 1806
16.1071(PORT1_PROFILE_ATTRIBUTES_REG53)—Offset 89E0h 1806
16.1072(PORT1_PROFILE_ATTRIBUTES_REG54)—Offset 89E4h 1807
16.1073(PORT1_PROFILE_ATTRIBUTES_REG55)—Offset 89E8h 1807

16.1074(PORT1_PROFILE_ATTRIBUTES_REG56)—Offset 89ECh 1808
16.1075(PORT1_PROFILE_ATTRIBUTES_REG57)—Offset 89F0h 1808
16.1076(PORT1_PROFILE_ATTRIBUTES_REG58)—Offset 89F4h 1809
16.1077(PORT1_PROFILE_ATTRIBUTES_REG59)—Offset 89F8h 1809
16.1078(PORT1_PROFILE_ATTRIBUTES_REG60)—Offset 89FCh 1810
16.1079(PORT1_PROFILE_ATTRIBUTES_REG61)—Offset 8A00h 1810
16.1080(PORT1_PROFILE_ATTRIBUTES_REG62)—Offset 8A04h 1811
16.1081(PORT1_PROFILE_ATTRIBUTES_REG63)—Offset 8A08h 1811
16.1082GLOBAL_TIME_SYNC_CAP_REG (GLOBAL_TIME_SYNC_CAP_REG)—Offset 8E10h .1812
16.1083GLOBAL_TIME_SYNC_CTRL_REG (GLOBAL_TIME_SYNC_CTRL_REG)—Offset 8E14h 1812
16.1084MICROFRAME_TIME_REG (MICROFRAME_TIME_REG)—Offset 8E18h 1812
16.1085GLOBAL_TIME_LOW_REG (GLOBAL_TIME_LOW_REG)—Offset 8E20h 1813
16.1086GLOBAL_TIME_HI_REG (GLOBAL_TIME_HI_REG)—Offset 8E24h 1813
16.1087Debug Status Capability Register (DEBUG_STATUS_CAPABILITY_REG)—Offset 8E58h .. 1814
16.1088Host Ctrl USB3 Soft Error Count Register 1 (HOST_CTRL_USB3_ERR_COUNT_REG1)—Offset 8E5Ch 1814
16.1089Host Ctrl USB3 Soft Error Count Register 2 (HOST_CTRL_USB3_ERR_COUNT_REG2)—Offset 8E60h 1814
16.1090Host Ctrl USB3 Soft Error Count Register 3 (HOST_CTRL_USB3_ERR_COUNT_REG3)—Offset 8E64h 1815
16.1091Host Ctrl USB3 Soft Error Count Register 4 (HOST_CTRL_USB3_ERR_COUNT_REG4)—Offset 8E68h 1815
16.1092Host Ctrl USB3 Soft Error Count Register 5 (HOST_CTRL_USB3_ERR_COUNT_REG5)—Offset 8E6Ch 1816
16.1093Host Ctrl USB3 Soft Error Count Register 6 (HOST_CTRL_USB3_ERR_COUNT_REG6)—Offset 8E70h 1816
16.1094Host Ctrl USB3 Soft Error Count Register 7 (HOST_CTRL_USB3_ERR_COUNT_REG7)—Offset 8E74h 1816
16.1095(GEN_REGRW1)—Offset B0h 1817
16.1096(GEN_REGRW2)—Offset B4h 1817
16.1097(GEN_REGRW3)—Offset B8h 1817
16.1098(GEN_REGRW4)—Offset BCh 1818
16.1099 (GEN_INPUT_REGRW)—Offset C0h 1818
16.1100(GEN_REGRW1)—Offset B0h 1819
16.1101(GEN_REGRW2)—Offset B4h 1820
16.1102(GEN_REGRW3)—Offset B8h 1820
16.1103(GEN_REGRW4)—Offset BCh 1820
16.1104 (GEN_INPUT_REGRW)—Offset C0h 1821
16.1105APBFC_U3PMU_CFG0 (APBFC_U3PMU_CFG0)—Offset 10F808h 1822
16.1106APBFC_U3PMU_CFG1 (APBFC_U3PMU_CFG1)—Offset 10F80Ch 1822
16.1107APBFC_U3PMU_CFG2 (APBFC_U3PMU_CFG2)—Offset 10F810h 1824
16.1108APBFC_U3PMU_CFG3 (APBFC_U3PMU_CFG3)—Offset 10F814h 1824
16.1109APBFC_U3PMU_CFG4 (APBFC_U3PMU_CFG4)—Offset 10F818h 1825
16.1110APBFC_U3PMU_CFG5 (APBFC_U3PMU_CFG5)—Offset 10F81Ch 1826
16.1111APBFC_U3PMU_CFG6 (APBFC_U3PMU_CFG6)—Offset 10F820h 1827
16.1112APBFC_D0I3C (APBFC_D0I3C)—Offset 10F830h 1827
16.1113GTXTHRCFG (GTXTHRCFG)—Offset C108h 1829
16.1114GRXTHRCFG (GRXTHRCFG)—Offset C10Ch 1830
16.1115GCTL (GCTL)—Offset C110h 1830
16.1116GPMSTS (GPMSTS)—Offset C114h 1831
16.1117GSTS (GSTS)—Offset C118h 1832
16.1118GUCTL1 (GUCTL1)—Offset C11Ch 1832
16.1119GSNPSID (GSNPSID)—Offset C120h 1833

16.1120GGPIO (GGPIO)—Offset C124h 1833
16.1121GUID (GUID)—Offset C128h 1834
16.1122GUCTL (GUCTL)—Offset C12Ch 1834
16.1123GBUSERRADDRLO (GBUSERRADDRLO)—Offset C130h 1835
16.1124GBUSERRADDRHI (GBUSERRADDRHI)—Offset C134h 1835
16.1125GPRTBIMAPLO (GPRTBIMAPLO)—Offset C138h 1835
16.1126GPRTBIMAPHI (GPRTBIMAPHI)—Offset C13Ch 1836
16.1127GHWPARAMS0 (GHWPARAMS0)—Offset C140h 1837
16.1128GHWPARAMS1 (GHWPARAMS1)—Offset C144h 1837
16.1129GHWPARAMS2 (GHWPARAMS2)—Offset C148h 1838
16.1130GHWPARAMS3 (GHWPARAMS3)—Offset C14Ch 1838
16.1131GHWPARAMS4 (GHWPARAMS4)—Offset C150h 1839
16.1132GHWPARAMS5 (GHWPARAMS5)—Offset C154h 1840
16.1133GHWPARAMS6 (GHWPARAMS6)—Offset C158h 1841
16.1134GHWPARAMS7 (GHWPARAMS7)—Offset C15Ch 1841
16.1135GDBGFIFOSPACE (GDBGFIFOSPACE)—Offset C160h 1842
16.1136GDBGLTSSM (GDBGLTSSM)—Offset C164h 1842
16.1137GDBGLNMCC (GDBGLNMCC)—Offset C168h 1843
16.1138GDBGBMU (GDBGBMU)—Offset C16Ch 1844
16.1139GDBGLSPMUX_DEV (GDBGLSPMUX_DEV)—Offset C170h 1844
16.1140GDBGLSP (GDBGLSP)—Offset C174h 1844
16.1141GDBGEPINFO0 (GDBGEPINFO0)—Offset C178h 1845
16.1142GDBGEPINFO1 (GDBGEPINFO1)—Offset C17Ch 1845
16.1143GPRTBIMAP_HSLO (GPRTBIMAP_HSLO)—Offset C180h 1845
16.1144GPRTBIMAP_HSHI (GPRTBIMAP_HSHI)—Offset C184h 1846
16.1145GPRTBIMAP_FSLO (GPRTBIMAP_FSLO)—Offset C188h 1847
16.1146GPRTBIMAP_FSHI (GPRTBIMAP_FSHI)—Offset C18Ch 1847
16.1147Reserved_94 (Reserved_94)—Offset C194h 1848
16.1148Reserved_98 (Reserved_98)—Offset C198h 1848
16.1149GUSB2PHYCFG_0 (GUSB2PHYCFG_0)—Offset C200h 1848
16.1150GUSB2I2CCTL_0 (GUSB2I2CCTL_0)—Offset C240h 1849
16.1151GUSB2PHYACC_ULPI_0 (GUSB2PHYACC_ULPI_0)—Offset C280h 1850
16.1152GUSB3PIPECTL_0 (GUSB3PIPECTL_0)—Offset C2C0h 1850
16.1153GTXFIFOSIZ0_0 (GTXFIFOSIZ0_0)—Offset C300h 1852
16.1154GTXFIFOSIZ1_0 (GTXFIFOSIZ1_0)—Offset C304h 1852
16.1155GTXFIFOSIZ2_0 (GTXFIFOSIZ2_0)—Offset C308h 1853
16.1156GTXFIFOSIZ3_0 (GTXFIFOSIZ3_0)—Offset C30Ch 1853
16.1157GTXFIFOSIZ4_0 (GTXFIFOSIZ4_0)—Offset C310h 1853
16.1158GTXFIFOSIZ5_0 (GTXFIFOSIZ5_0)—Offset C314h 1854
16.1159GTXFIFOSIZ6_0 (GTXFIFOSIZ6_0)—Offset C318h 1854
16.1160GTXFIFOSIZ7_0 (GTXFIFOSIZ7_0)—Offset C31Ch 1854
16.1161GTXFIFOSIZ8_0 (GTXFIFOSIZ8_0)—Offset C320h 1855
16.1162GTXFIFOSIZ9_0 (GTXFIFOSIZ9_0)—Offset C324h 1855
16.1163GTXFIFOSIZ10_0 (GTXFIFOSIZ10_0)—Offset C328h 1855
16.1164GTXFIFOSIZ11_0 (GTXFIFOSIZ11_0)—Offset C32Ch 1856
16.1165GTXFIFOSIZ12_0 (GTXFIFOSIZ12_0)—Offset C330h 1856
16.1166GTXFIFOSIZ13_0 (GTXFIFOSIZ13_0)—Offset C334h 1856
16.1167GTXFIFOSIZ14_0 (GTXFIFOSIZ14_0)—Offset C338h 1857
16.1168GTXFIFOSIZ15_0 (GTXFIFOSIZ15_0)—Offset C33Ch 1857
16.1169GRXFIFOSIZ0_0 (GRXFIFOSIZ0_0)—Offset C380h 1857
16.1170GEVNTADRLO_0 (GEVNTADRLO_0)—Offset C400h 1858
16.1171GEVNTADRHI_0 (GEVNTADRHI_0)—Offset C404h 1858
16.1172GEVNTSIZ_0 (GEVNTSIZ_0)—Offset C408h 1858
16.1173GEVNTCOUNT_0 (GEVNTCOUNT_0)—Offset C40Ch 1859
16.1174GHWPARAMS8 (GHWPARAMS8)—Offset C600h 1859

16.1175GTXFIFOPRIDEV (GTXFIFOPRIDEV)—Offset C610h 1859
16.1176GFLADJ (GFLADJ)—Offset C630h.....1860
16.1177RID - Bridge revision ID Register (RID)—Offset 30h.....1860
16.1178GEN_REGRW1 - General Purpose register (GEN_REGRW1)—Offset 600h.....1861
16.1179GEN_REGRW2 - General Purpose register (GEN_REGRW2)—Offset 604h.....1861
16.1180GEN_REGRW3 - General Purpose register (GEN_REGRW3)—Offset 608h.....1861
16.1181GEN_REGRW4 - General Purpose register (GEN_REGRW4)—Offset 60Ch 1862
16.1182(DEVVENDID)—Offset 0h 1862
16.1183(STATUSCOMMAND)—Offset 4h 1862
16.1184(REVCLASSCODE)—Offset 8h.....1863
16.1185(CLLATHEADERBIST)—Offset Ch 1864
16.1186(BAR)—Offset 10h.....1864
16.1187(BAR1)—Offset 18h 1865
16.1188(SUBSYSTEMID)—Offset 2Ch 1865
16.1189(EXPANSION_ROM_BASEADDR)—Offset 30h.....1866
16.1190(CAPABILITYPTR)—Offset 34h.....1866
16.1191(INTERRUPTREG)—Offset 3Ch.....1866
16.1192(POWERCAPID)—Offset 80h 1867
16.1193(PMECTRLSTATUS)—Offset 84h.....1867
16.1194(PCIDEVIDLE_CAP_RECORD)—Offset 90h.....1868
16.1195(DEVID_VEND_SPECIFIC_REG)—Offset 94h.....1868
16.1196(D0I3_CONTROL_SW_LTR_MMIO_REG)—Offset 98h 1869
16.1197(DEVICE_IDLE_POINTER_REG)—Offset 9Ch.....1869
16.1198(D0I3_MAX_POW_LAT_PG_CONFIG)—Offset A0h 1870
16.1199(MANID)—Offset F8h 1870
16.1200(DEVVENDID)—Offset 0h 1871
16.1201(STATUSCOMMAND)—Offset 4h 1871
16.1202(REVCLASSCODE)—Offset 8h.....1872
16.1203(CLLATHEADERBIST)—Offset Ch 1872
16.1204(BAR)—Offset 10h.....1873
16.1205(BAR1)—Offset 18h 1873
16.1206(SUBSYSTEMID)—Offset 2Ch 1874
16.1207(EXPANSION_ROM_BASEADDR)—Offset 30h.....1874
16.1208(CAPABILITYPTR)—Offset 34h.....1874
16.1209(INTERRUPTREG)—Offset 3Ch.....1875
16.1210(POWERCAPID)—Offset 80h 1875
16.1211(PMECTRLSTATUS)—Offset 84h.....1876
16.1212(PCIDEVIDLE_CAP_RECORD)—Offset 90h.....1876
16.1213(DEVID_VEND_SPECIFIC_REG)—Offset 94h.....1877
16.1214(D0I3_CONTROL_SW_LTR_MMIO_REG)—Offset 98h 1877
16.1215(DEVICE_IDLE_POINTER_REG)—Offset 9Ch.....1878
16.1216(D0I3_MAX_POW_LAT_PG_CONFIG)—Offset A0h 1878
16.1217(MANID)—Offset F8h 1879

17 USB-Device (xDCI) Registers 1880

17.1 DEVVENDID — Offset 0h 1880
17.2 STATUSCOMMAND — Offset 4h 1880
17.3 REVCLASSCODE — Offset 8h 1881
17.4 CLLATHEADERBIST — Offset Ch 1881
17.5 BAR — Offset 10h.....1882
17.6 BAR1 — Offset 18h.....1882
17.7 SUBSYSTEMID — Offset 2Ch.....1883
17.8 EXPANSION_ROM_BASEADDR — Offset 30h 1883
17.9 CAPABILITYPTR — Offset 34h 1883
17.10 INTERRUPTREG — Offset 3Ch 1884

17.11 POWERCAPID — Offset 80h 1884
17.12 PMECTRLSTATUS — Offset 84h 1885
17.13 PCIDEVIDLE_CAP_RECORD — Offset 90h 1885
17.14 DEVID_VEND_SPECIFIC_REG — Offset 94h 1886
17.15 D0I3_CONTROL_SW_LTR_MMIO_REG — Offset 98h 1886
17.16 DEVICE_IDLE_POINTER_REG — Offset 9Ch 1886
17.17 D0I3_MAX_POW_LAT_PG_CONFIG — Offset A0h 1887
17.18 GEN_REGRW1 — Offset B0h 1887
17.19 GEN_REGRW2 — Offset B4h 1888
17.20 GEN_REGRW3 — Offset B8h 1888
17.21 GEN_REGRW4 — Offset BCh 1888
17.22 GEN_INPUT_REGRW — Offset C0h 1889
17.23 MANID — Offset F8h 1890

18 PCI Express* Controller Registers 1891

18.1 ID — Offset 0h 1891
18.2 Device Command; Primary Status (CMD_PSTS) — Offset 4h 1891
18.3 Revision ID;Class Code (RID_CC) — Offset 8h 1893
18.4 Cache Line Size; Primary Latency Timer; Header Type (CLS_PLT_HTYPE) — Offset Ch ... 1893
18.5 Bus Numbers; Secondary Latency Timer (BNUM_SLT) — Offset 18h 1894
18.6 I/O Base and Limit; Secondary Status (IOBL_SSTS) — Offset 1Ch 1894
18.7 Memory Base and Limit (MBL) — Offset 20h 1895
18.8 Prefetchable Memory Base and Limit (PMBL) — Offset 24h 1896
18.9 Prefetchable Memory Base Upper 32 Bits (PMBU32) — Offset 28h 1896
18.10 Prefetchable Memory Limit Upper 32 Bits (PMLU32) — Offset 2Ch 1896
18.11 Capabilities List Pointer (CAPP) — Offset 34h 1897
18.12 Interrupt Information; Bridge Control (INTR_BCTRL) — Offset 3Ch 1897
18.13 Capabilities List; PCI Express Capabilities (CLIST_XCAP) — Offset 40h 1898
18.14 Device Capabilities (DCAP) — Offset 44h 1899
18.15 Device Control; Device Status (DCTL_DSTS) — Offset 48h 1900
18.16 Link Capabilities (LCAP) — Offset 4Ch 1901
18.17 Link Control; Link Status (LCTL_LSTS) — Offset 50h 1902
18.18 Slot Capabilities (SLCAP) — Offset 54h 1904
18.19 Slot Control; Slot Status (SLCTL_SLSTS) — Offset 58h 1905
18.20 Root Control (RCTL) — Offset 5Ch 1906
18.21 Root Status (RSTS) — Offset 60h 1907
18.22 Device Capabilities 2 (DCAP2) — Offset 64h 1907
18.23 Device Control 2; Device Status 2 (DCTL2_DSTS2) — Offset 68h 1908
18.24 Link Capabilities 2 (LCAP2) — Offset 6Ch 1909
18.25 Link Control 2; Link Status 2 (LCTL2_LSTS2) — Offset 70h 1910
18.26 Slot Capabilities 2 (SLCAP2) — Offset 74h 1911
18.27 Slot Control 2; Slot Status 2 (SLCTL2_SLSTS2) — Offset 78h 1912
18.28 Message Signaled Interrupt Identifiers; Message Signaled Interrupt Message Control (MID_MC) — Offset 80h 1912
18.29 Message Signaled Interrupt Message Address (MA) — Offset 84h 1913
18.30 Message Signaled Interrupt Message Data (MD) — Offset 88h 1913
18.31 Subsystem Vendor Capability (SVCAP) — Offset 90h 1913
18.32 Subsystem Vendor IDs (SVID) — Offset 94h 1914
18.33 Power Management Capability; PCI Power Management Capabilities (PMCAP_PMC) — Offset A0h 1914
18.34 PCI Power Management Control And Status (PMCS) — Offset A4h 1915
18.35 Advanced Error Extended Reporting Capability Header (AECH) — Offset 100h 1916
18.36 Uncorrectable Error Status (UES) — Offset 104h 1916
18.37 Uncorrectable Error Mask (UEM) — Offset 108h 1917

18.38 Uncorrectable Error Severity (UEV) — Offset 10Ch 1918
18.39 Correctable Error Status (CES) — Offset 110h 1919
18.40 Correctable Error Mask (CEM) — Offset 114h 1920
18.41 Advanced Error Capabilities and Control (AECC) — Offset 118h 1920
18.42 Header Log DW1 (HL_DW1) — Offset 11Ch 1921
18.43 Header Log DW2 (HL_DW2) — Offset 120h 1921
18.44 Header Log DW3 (HL_DW3) — Offset 124h 1921
18.45 Header Log DW4 (HL_DW4) — Offset 128h 1922
18.46 Root Error Command (REC) — Offset 12Ch 1922
18.47 Root Error Status (RES) — Offset 130h 1923
18.48 Error Source Identification (ESID) — Offset 134h 1923
18.49 ACS Extended Capability Header (ACSECH) — Offset 140h 1924
18.50 ACS Capability Register; ACS Control Register (ACSCAPR_ACSCTLR) — Offset 144h 1924
18.51 PTM Extended Capability Header (PTMECH) — Offset 150h 1925
18.52 PTM Capability Register (PTMCAPR) — Offset 154h 1925
18.53 PTM Control Register (PTMCTLR) — Offset 158h 1926
18.54 L1 Sub-States Extended Capability Header (L1SECH) — Offset 200h 1926
18.55 L1 Sub-States Capabilities (L1SCAP) — Offset 204h 1927
18.56 L1 Sub-States Control 1 (L1SCTL1) — Offset 208h 1928
18.57 L1 Sub-States Control 2 (L1SCTL2) — Offset 20Ch 1929
18.58 Secondary PCI Express Extended Capability Header (SPEECH) — Offset 220h 1929
18.59 Link Control 3 (LCTL3) — Offset 224h 1930
18.60 Lane Error Status (LES) — Offset 228h 1930
18.61 Lane 0 and Lane 1 Equalization Control (L01EC) — Offset 22Ch 1931
18.62 Lane 2 and Lane 3 Equalization Control (L23EC) — Offset 230h 1932
18.63 PCI Express Replay Timer Policy 1 (PCIERTP1) — Offset 300h 1933
18.64 PCI Express Replay Timer Policy 2 (PCIERTP2) — Offset 304h 1934
18.65 PCI Express Status 1 (PCIESTS1) — Offset 328h 1935
18.66 PCI Express Status 2 (PCIESTS2) — Offset 32Ch 1936
18.67 PCI Express Compliance Measurement Mode (CMM) Port Control (PCIECMMPC) — Offset 330h 1937
18.68 PCI Express Compliance Measurement Mode Symbol Buffer (PCIECMMSB) — Offset 334h 1938
18.69 PTM Propagation Delay (PTMPD) — Offset 390h 1939
18.70 PTM Lower Local Master Time (PTMLLMT) — Offset 394h 1939
18.71 PTM Upper Local Master Time (PTMULMT) — Offset 398h 1939
18.72 PTM Pipe Stage Delay Configuration 1 (PTMPSDC1) — Offset 39Ch 1940
18.73 PTM Pipe Stage Delay Configuration 2 (PTMPSDC2) — Offset 3A0h 1940
18.74 PTM Pipe Stage Delay Configuration 3 (PTMPSDC3) — Offset 3A4h 1941
18.75 PTM Pipe Stage Delay Configuration 4 (PTMPSDC4) — Offset 3A8h 1941
18.76 PTM Pipe Stage Delay Configuration 5 (PTMPSDC5) — Offset 3ACh 1942
18.77 PTM Extended Config (PTMECFG) — Offset 3B0h 1942
18.78 PTM Lower T2 Time Stamp (PTMLT2TSTMP) — Offset 3B4h 1943
18.79 PTM Upper T2 Time Stamp (PTMUT2TSTMP) — Offset 3B8h 1944
18.80 Strap and Fuse Configuration 2 (STRPFUSECFG2) — Offset 414h 1944
18.81 Thermal and Power Throttling (TNPT) — Offset 418h 1945
18.82 Dynamic Lane Switch (DYNLNSW) — Offset 41Ch 1946
18.83 Power Control Enable (PCE) — Offset 428h 1946
18.84 PGCB Control1 (PGCBCTL1) — Offset 42Ch 1947
18.85 PGCB Control2 (PGCBCTL2) — Offset 430h 1948
18.86 Equalization Configuration 1 (EQCFG1) — Offset 450h 1948
18.87 Remote Transmitter Preset Coefficient List 1 (RTPCL1) — Offset 454h 1951
18.88 Remote Transmitter Preset Coefficient List 2 (RTPCL2) — Offset 458h 1953
18.89 Remote Transmitter Preset Coefficient List 3 (RTPCL3) — Offset 45Ch 1954
18.90 Remote Transmitter Preset Coefficient List 4 (RTPCL4) — Offset 460h 1955

18.91 Figure Of Merit Status (FOMS) — Offset 464h 1957
18.92 Hardware Autonomous Equalization Control (HAEQ) — Offset 468h 1957
18.93 Local Transmitter Coefficient Override 1 (LTCO1) — Offset 470h 1958
18.94 Local Transmitter Coefficient Override 2 (LTCO2) — Offset 474h 1959
18.95 GEN3 L0s Control (G3L0SCTL) — Offset 478h 1960
18.96 Equalization Configuration 2 (EQCFG2) — Offset 47Ch 1961
18.97 Monitor Mux (MM) — Offset 480h 1962
18.98 Lane0 P0 and P1 Preset-Coefficient Mapping (L0P0P1PCM) — Offset 500h 1963
18.99 Lane0 P1, P2 and P3 Preset-Coefficient Mapping (L0P1P2P3PCM) — Offset 504h 1963
18.100 Lane0 P3 and P4 Preset-Coefficient Mapping (L0P3P4PCM) — Offset 508h 1964
18.101 Lane0 P5 and P6 Preset-Coefficient Mapping (L0P5P6PCM) — Offset 50Ch 1965
18.102 Lane0 P6, P7 and P8 Preset-Coefficient Mapping (L0P6P7P8PCM) — Offset 510h 1965
18.103 Lane0 P8 and P9 Preset-Coefficient Mapping (L0P8P9PCM) — Offset 514h 1966
18.104 Lane0 P10 Preset-Coefficient Mapping (L0P10PCM) — Offset 518h 1967
18.105 Lane0 LF and FS (L0LFFS) — Offset 51Ch 1967
18.106 Lane1 P0 and P1 Preset-Coefficient Mapping (L1P0P1PCM) — Offset 520h 1968
18.107 Lane1 P1, P2 and P3 Preset-Coefficient Mapping (L1P1P2P3PCM) — Offset 524h 1969
18.108 Lane1 P3 and P4 Preset-Coefficient Mapping (L1P3P4PCM) — Offset 528h 1969
18.109 Lane1 P5 and P6 Preset-Coefficient Mapping (L1P5P6PCM) — Offset 52Ch 1970
18.110 Lane1 P6, P7 and P8 Preset-Coefficient Mapping (L1P6P7P8PCM) — Offset 530h 1971
18.111 Lane1 P8 and P9 Preset-Coefficient Mapping (L1P8P9PCM) — Offset 534h 1971
18.112 Lane1 P10 Preset-Coefficient Mapping (L1P10PCM) — Offset 538h 1972
18.113 Lane1 LF and FS (L1LFFS) — Offset 53Ch 1973
18.114 Lane2 P0 and P1 Preset-Coefficient Mapping (L2P0P1PCM) — Offset 540h 1973
18.115 Lane2 P1, P2 and P3 Preset-Coefficient Mapping (L2P1P2P3PCM) — Offset 544h 1974
18.116 Lane2 P3 and P4 Preset-Coefficient Mapping (L2P3P4PCM) — Offset 548h 1975
18.117 Lane2 P5 and P6 Preset-Coefficient Mapping (L2P5P6PCM) — Offset 54Ch 1975
18.118 Lane2 P6, P7 and P8 Preset-Coefficient Mapping (L2P6P7P8PCM) — Offset 550h 1976
18.119 Lane2 P8 and P9 Preset-Coefficient Mapping (L2P8P9PCM) — Offset 554h 1977
18.120 Lane2 P10 Preset-Coefficient Mapping (L2P10PCM) — Offset 558h 1977
18.121 Lane2 LF and FS (L2LFFS) — Offset 55Ch 1978
18.122 Lane3 P0 and P1 Preset-Coefficient Mapping (L3P0P1PCM) — Offset 560h 1979
18.123 Lane3 P1, P2 and P3 Preset-Coefficient Mapping (L3P1P2P3PCM) — Offset 564h 1979
18.124 Lane3 P3 and P4 Preset-Coefficient Mapping (L3P3P4PCM) — Offset 568h 1980
18.125 Lane3 P5 and P6 Preset-Coefficient Mapping (L3P5P6PCM) — Offset 56Ch 1981
18.126 Lane3 P6, P7 and P8 Preset-Coefficient Mapping (L3P6P7P8PCM) — Offset 570h 1981
18.127 Lane3 P8 and P9 Preset-Coefficient Mapping (L3P8P9PCM) — Offset 574h 1982
18.128 Lane3 P10 Preset-Coefficient Mapping (L3P10PCM) — Offset 578h 1983
18.129 Lane3 LF and FS (L3LFFS) — Offset 57Ch 1983
18.130 Common Control (CC)—Offset 0h 1984
18.131 Device Reference Clock Request Mapping 1 (DRCRM1)—Offset 100h 1986
18.132 Device Reference Clock Request Mapping 2 (DRCRM2)—Offset 104h 1993
18.133 Device Reference Clock Request Mapping 3 (DRCRM3)—Offset 108h 2001

19 SATA Registers 2006

19.1 ID — Offset 0h 2006
19.2 Command (CMD) — Offset 4h 2006
19.3 Device Status (STS) — Offset 6h 2007
19.4 Revision ID (RID) — Offset 8h 2008
19.5 Programming Interface (PI) — Offset 9h 2008
19.6 Class Code (CC) — Offset Ah 2008
19.7 Cache Line Size (CLS) — Offset Ch 2009
19.8 Master Latency Timer (MLT) — Offset Dh 2009
19.9 Header Type (HTYPE) — Offset Eh 2009
19.10 MSI-X Table Base Address (MXTBA) — Offset 10h 2010

19.11 MXP Base Address (MXPBA) — Offset 14h 2010
19.12 SCMDBA — Offset 18h 2011
19.13 SCTLBA — Offset 1Ch 2011
19.14 AHCI Index Data Pair Base Address (AIDPBA) — Offset 20h 2012
19.15 AHCI Base Address (ABAR) — Offset 24h 2012
19.16 Sub System Identifiers (SS) — Offset 2Ch 2013
19.17 CAP — Offset 34h 2013
19.18 Interrupt Information (INTR) — Offset 3Ch 2014
19.19 PCI Power Management Capability ID (PID) — Offset 70h 2014
19.20 PC — Offset 72h 2014
19.21 PCI Power Management Control and Status (PMCS) — Offset 74h 2015
19.22 Message Signaled Interrupt Identifier (MID) — Offset 80h 2016
19.23 Message Signalled Interrupt Message Control (MC) — Offset 82h 2016
19.24 Message Signalled Interrupt Message Address (MA) — Offset 84h 2017
19.25 Message Signalled Interrupt Message Data (MD) — Offset 88h 2017
19.26 Port Mapping Register (MAP) — Offset 90h 2017
19.27 Port Control and Status (PCS) — Offset 94h 2019
19.28 SATA General Configuration (SATAGC) — Offset 9Ch 2020
19.29 SATA Initialization Register Index (SIRI) — Offset A0h 2022
19.30 SATA Initialization Register Data (SIRD) — Offset A4h 2022
19.31 Serial ATA Capability Register 0 (SATACR0) — Offset A8h 2022
19.32 Serial ATA Capability Register 1 (SATACR1) — Offset ACh 2023
19.33 FLR Capability ID (FLRCID) — Offset B0h 2023
19.34 FLR Capability Length and Version (FLRCAP) — Offset B2h 2024
19.35 FLR Control (FLRCTL) — Offset B4h 2024
19.36 Scratch Pad (SP) — Offset C0h 2025
19.37 MSI-X Identifiers (MXID) — Offset D0h 2025
19.38 MSI-X Message Control (MXC) — Offset D2h 2025
19.39 MSI-X Table Offset / Table BIR (MXT) — Offset D4h 2026
19.40 MSI-X PBA Offset / PBA BIR (MXP) — Offset D8h 2026
19.41 BIST FIS Control/Status (BFCS) — Offset E0h 2027
19.42 BIST FIS Transmit Data 1 (BFTD1) — Offset E4h 2028
19.43 BIST FIS Transmit Data 2 (BFTD2) — Offset E8h 2029
19.44 Manufacturing ID (MFID) — Offset F8h 2029
19.45 AHCI Index Register (INDEX)—Offset 10h 2030
19.46 AHCI Data Register (DATA)—Offset 14h 2030
19.47 HBA Capabilities (GHC_CAP)—Offset 0h 2030
19.48 Global HBA Control (GHC)—Offset 4h 2033
19.49 Interrupt Status Register (IS)—Offset 8h 2034
19.50 Ports Implemented (GHC_PI)—Offset Ch 2035
19.51 AHCI Version (VS)—Offset 10h 2036
19.52 Enclosure Management Location (EM_LOC)—Offset 1Ch 2037
19.53 Enclosure Management Control (EM_CTL)—Offset 20h 2037
19.54 HBA Capabilities Extended (GHC_CAP2)—Offset 24h 2039
19.55 Vendor Specific (VSP)—Offset A0h 2040
19.56 Vendor-Specific Capabilities Register (VS_CAP)—Offset A4h 2040
19.57 Remapping Under NVMe (RUN)—Offset A8h 2041
19.58 RAID Platform ID (RPID)—Offset C0h 2042
19.59 Premium Feature Block (PFB)—Offset C4h 2042
19.60 SW Feature Mask (SFM)—Offset C8h 2043
19.61 Port [0-7] Command List Base Address (PxCLB0)—Offset 100h 2044
19.62 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU0)—Offset 104h 2045
19.63 Port [0-7] FIS Base Address (PxFB0)—Offset 108h 2045
19.64 Port [0-7] FIS Base Address Upper 32-bits (PxFBU0)—Offset 10Ch 2045
19.65 Port [0-7] Interrupt Status (PxIS0)—Offset 110h 2046

19.66 Port [0-7] Interrupt Enable (PxIE0)—Offset 114h 2047
19.67 Port [0-7] Command (PxCMD0)—Offset 118h 2049
19.68 Port [0-7] Task File Data (PxTFD0)—Offset 120h 2051
19.69 Port [0-7] Signature (PxSIG0)—Offset 124h 2052
19.70 Port [0-7] Serial ATA Status (PxSSTS0)—Offset 128h 2052
19.71 Port [0-7] Serial ATA Control (PxSCTL0)—Offset 12Ch 2053
19.72 Port [0-7] Serial ATA Error (PxSERR0)—Offset 130h 2053
19.73 Port [0-7] Serial ATA Active (PxSACT0)—Offset 134h 2054
19.74 Port [0-7] Commands Issued (PxCI0)—Offset 138h 2054
19.75 Port [0-7] SNotification (PxSNTF0)—Offset 13Ch 2055
19.76 Port [0-7] Device Sleep (PxDEVSLP0)—Offset 144h 2055
19.77 Port [0-7] Command List Base Address (PxCLB1)—Offset 180h 2057
19.78 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU1)—Offset 184h 2057
19.79 Port [0-7] FIS Base Address (PxFB1)—Offset 188h 2057
19.80 Port [0-7] FIS Base Address Upper 32-bits (PxFBU1)—Offset 18Ch 2058
19.81 Port [0-7] Interrupt Status (PxIS1)—Offset 190h 2058
19.82 Port [0-7] Interrupt Enable (PxIE1)—Offset 194h 2060
19.83 Port [0-7] Command (PxCMD1)—Offset 198h 2061
19.84 Port [0-7] Task File Data (PxTFD1)—Offset 1A0h 2064
19.85 Port [0-7] Signature (PxSIG1)—Offset 1A4h 2065
19.86 Port [0-7] Serial ATA Status (PxSSTS1)—Offset 1A8h 2065
19.87 Port [0-7] Serial ATA Control (PxSCTL1)—Offset 1ACh 2066
19.88 Port [0-7] Serial ATA Error (PxSERR1)—Offset 1B0h 2066
19.89 Port [0-7] Serial ATA Active (PxSACT1)—Offset 1B4h 2067
19.90 Port [0-7] Commands Issued (PxCI1)—Offset 1B8h 2067
19.91 Port [0-7] SNotification (PxSNTF1)—Offset 1BCh 2067
19.92 Port [0-7] Device Sleep (PxDEVSLP1)—Offset 1C4h 2068
19.93 Enclosure Management Message Format (EM_MF)—Offset 580h 2069
19.94 Enclosure Management LED (EM_LED)—Offset 584h 2070
19.95 MSI-X Pending Bit Array QW 0 (MXPQW0_DW0)—Offset 0h 2071
19.96 MSI-X Pending Bit Array QW 1 (MXPQW0_DW1)—Offset 4h 2072
19.97 MSI-X Table Entries 0 Message Lower Address (MXTE0MLA)—Offset 0h 2072
19.98 MSI-X Table Entries 0 Message Upper Address (MXTE0MUA)—Offset 4h 2073
19.99 MSI-X Table Entries 0 Message Data (MXTE0MD)—Offset 8h 2073
19.100 MSI-X Table Entries 0 Vector Control (MXTE0VC)—Offset Ch 2073

20 I/O Sensor Hub (ISH) Registers 2075

20.1 DEVICEVENDORID - Device ID and Vendor ID Register (DEVVENDID) — Offset 0h . 2075
20.2 STATUSCOMMAND — Offset 4h 2075
20.3 REVCLASSCODE — Offset 8h 2076
20.4 CLLATHEADERBIST — Offset Ch 2076
20.5 BAR — Offset 10h 2077
20.6 BAR -Base Address Register High (BAR_HIGH) — Offset 14h 2077
20.7 BAR1 — Offset 18h 2078
20.8 BAR1 -Base Address Register1 High (BAR1_HIGH) — Offset 1Ch 2078
20.9 SUBSYSTEMID — Offset 2Ch 2078
20.10 EXPANSION ROM base address (EXPANSION_ROM_BASEADDR) — Offset 30h 2079
20.11 CAPABILITYPTR — Offset 34h 2079
20.12 INTERRUPTREG — Offset 3Ch 2079
20.13 POWERCAPID — Offset 80h 2080
20.14 PMECTRLSTATUS — Offset 84h 2080
20.15 PCI DEVICE IDLE CAPABILITY RECORD (PCIDEVIDLE_CAP_RECORD) — Offset 90h. 2081
20.16 DEVID VENDOR SPECIFIC REG (DEVID_VEND_SPECIFIC_REG) — Offset 94h 2082
20.17 D0I3_CONTROL_SW_LTR_MMIO_REG — Offset 98h 2082
20.18 DEVICE_IDLE_POINTER_REG — Offset 9Ch 2082

20.19 D0I3_MAX_POW_LAT_PG_CONFIG — Offset A0h 2083
20.20 GEN_REGRW1 - General Purpose Read Write Register1 (GEN_REGRW1) — Offset B0h 2083
20.21 GEN_REGRW2 — Offset B4h 2084
20.22 GEN_REGRW3 — Offset B8h 2084
20.23 GEN_REGRW4 — Offset BCh 2084
20.24 GEN_INPUT_REG — Offset C0h 2085
20.25 MANID — Offset F8h 2085
20.26 Peripheral Interrupt Status (PISR)—Offset 1000h 2085
20.27 Peripheral Interrupt Mask (PIMR)—Offset 1004h 2087
20.28 Host Peripheral Interrupt Mask (HOST_PIMR)—Offset 1008h 2089
20.29 HOST Peripheral Interrupt Status (HOST_PISR)—Offset 100Ch 2089
20.30 Reserved (RESERVED2)—Offset 1010h 2090
20.31 Reserved (RESERVED3)—Offset 1014h 2090
20.32 Reserved (RESERVED4)—Offset 1018h 2090
20.33 Reserved (RESERVED5)—Offset 101Ch 2091
20.34 Reserved (RESERVED6)—Offset 1020h 2091
20.35 Reserved (RESERVED7)—Offset 1024h 2091
20.36 Reserved (RESERVED8)—Offset 1028h 2092
20.37 Reserved (RESERVED9)—Offset 102Ch 2092
20.38 Reserved (RESERVED10)—Offset 1030h 2092
20.39 ISH Host Firmware Status (ISH_HOST_FWSTS)—Offset 1034h 2093
20.40 Host Communication (HOST_COMM)—Offset 1038h 2093
20.41 ISH SEC Firmware Status (ISH_SEC_FWSTS)—Offset 103Ch 2093
20.42 SEC Communication (SEC_COMM)—Offset 1040h 2094
20.43 ISH Reset (ISH_RST)—Offset 1044h 2094
20.44 Inbound Doorbell Host to ISH (HOST2ISH_DOORBELL)—Offset 1048h 2094
20.45 Inbound Doorbell PMC To ISH (PMC2ISH_DOORBELL)—Offset 104Ch 2095
20.46 Inbound Doorbell SEC to ISH (SEC2ISH_DOORBELL)—Offset 1050h 2095
20.47 Outbound Doorbell ISH To Host (ISH2HOST_DOORBELL)—Offset 1054h 2096
20.48 Outbound Doorbell ISH to PMC (ISH2PMC_DOORBELL)—Offset 1058h 2096
20.49 Outbound Doorbell ISH to SEC (ISH2SEC_DOORBELL)—Offset 105Ch 2096
20.50 Outbound Inter Processor Messages 1 ISH to Host (ISH2HOST_MSG1)—Offset 1060h 2097
20.51 Outbound Inter Processor Messages 2 ISH to Host (ISH2HOST_MSG2)—Offset 1064h 2097
20.52 Outbound Inter Processor Messages 3 ISH to Host (ISH2HOST_MSG3)—Offset 1068h 2098
20.53 Outbound Inter Processor Messages 4 ISH to Host (ISH2HOST_MSG4)—Offset 106Ch 2098
20.54 Outbound Inter Processor Messages 5 ISH to Host (ISH2HOST_MSG5)—Offset 1070h 2098
20.55 Outbound Inter Processor Messages 6 ISH to Host (ISH2HOST_MSG6)—Offset 1074h 2099
20.56 Outbound Inter Processor Messages 7 ISH to Host (ISH2HOST_MSG7)—Offset 1078h 2099
20.57 Outbound Inter Processor Messages 8 ISH to Host (ISH2HOST_MSG8)—Offset 107Ch 2100
20.58 Outbound Inter Processor Messages 9 ISH to Host (ISH2HOST_MSG9)—Offset 1080h 2100
20.59 Outbound Inter Processor Messages 10 ISH to Host (ISH2HOST_MSG10)—Offset 1084h. 2100
20.60 Outbound Inter Processor Messages 11 ISH to Host (ISH2HOST_MSG11)—Offset 1088h. 2101
20.61 Outbound Inter Processor Messages 12 ISH to Host (ISH2HOST_MSG12)—Offset 108Ch. 2101

20.62 Outbound Inter Processor Messages 13 ISH to Host (ISH2HOST_MSG13)—Offset 1090h 2102
20.63 Outbound Inter Processor Messages 14 ISH to Host (ISH2HOST_MSG14)—Offset 1094h 2102
20.64 Outbound Inter Processor Messages 15 ISH to Host (ISH2HOST_MSG15)—Offset 1098h 2102
20.65 Outbound Inter Processor Messages 16 ISH to Host (ISH2HOST_MSG16)—Offset 109Ch 2103
20.66 Outbound Inter Processor Messages 17 ISH to Host (ISH2HOST_MSG17)—Offset 10A0h 2103
20.67 Outbound Inter Processor Messages 18 ISH to Host (ISH2HOST_MSG18)—Offset 10A4h 2104
20.68 Outbound Inter Processor Messages 19 ISH to Host (ISH2HOST_MSG19)—Offset 10A8h 2104
20.69 Outbound Inter Processor Messages 20 ISH to Host (ISH2HOST_MSG20)—Offset 10ACh 2104
20.70 Outbound Inter Processor Messages 21 ISH to Host (ISH2HOST_MSG21)—Offset 10B0h 2105
20.71 Outbound Inter Processor Messages 22 ISH to Host (ISH2HOST_MSG22)—Offset 10B4h 2105
20.72 Outbound Inter Processor Messages 23 ISH to Host (ISH2HOST_MSG23)—Offset 10B8h 2106
20.73 Outbound Inter Processor Messages 24 ISH to Host (ISH2HOST_MSG24)—Offset 10BCh 2106
20.74 Outbound Inter Processor Messages 25 ISH to Host (ISH2HOST_MSG25)—Offset 10C0h 2106
20.75 Outbound Inter Processor Messages 26 ISH to Host (ISH2HOST_MSG26)—Offset 10C4h 2107
20.76 Outbound Inter Processor Messages 27 ISH to Host (ISH2HOST_MSG27)—Offset 10C8h 2107
20.77 Outbound Inter Processor Messages 28 ISH to Host (ISH2HOST_MSG28)—Offset 10CCh 2108
20.78 Outbound Inter Processor Messages 29 ISH to Host (ISH2HOST_MSG29)—Offset 10D0h 2108
20.79 Outbound Inter Processor Messages 30 ISH to Host (ISH2HOST_MSG30)—Offset 10D4h 2108
20.80 Outbound Inter Processor Messages 31 ISH to Host (ISH2HOST_MSG31)—Offset 10D8h 2109
20.81 Outbound Inter Processor Messages 32 ISH to Host (ISH2HOST_MSG32)—Offset 10DCh 2109
20.82 Inbound Inter Processor Messages 1 Host to ISH (HOST2ISH_MSG1)—Offset 10E0h..... 2110
20.83 Inbound Inter Processor Messages 2 Host to ISH (HOST2ISH_MSG2)—Offset 10E4h..... 2110
20.84 Inbound Inter Processor Messages 3 Host to ISH (HOST2ISH_MSG3)—Offset 10E8h..... 2110
20.85 Inbound Inter Processor Messages 4 Host to ISH (HOST2ISH_MSG4)—Offset 10ECh..... 2111
20.86 Inbound Inter Processor Messages 5 Host to ISH (HOST2ISH_MSG5)—Offset 10F0h2111
20.87 Inbound Inter Processor Messages 6 Host to ISH (HOST2ISH_MSG6)—Offset 10F4h2112
20.88 Inbound Inter Processor Messages 7 Host to ISH (HOST2ISH_MSG7)—Offset 10F8h2112
20.89 Inbound Inter Processor Messages 8 Host to ISH (HOST2ISH_MSG8)—Offset 10FCh..... 2112
20.90 Inbound Inter Processor Messages 9 Host to ISH (HOST2ISH_MSG9)—Offset 1100h..... 2113
20.91 Inbound Inter Processor Messages 10 Host to ISH (HOST2ISH_MSG10)—Offset 1104h .. 2113

20.92 Inbound Inter Processor Messages 11 Host to ISH (HOST2ISH_MSG11)—Offset 1108h ... 2114
20.93 Inbound Inter Processor Messages 12 Host to ISH (HOST2ISH_MSG12)—Offset 110Ch ... 2114
20.94 Inbound Inter Processor Messages 13 Host to ISH (HOST2ISH_MSG13)—Offset 1110h ... 2114
20.95 Inbound Inter Processor Messages 14 Host to ISH (HOST2ISH_MSG14)—Offset 1114h ... 2115
20.96 Inbound Inter Processor Messages 15 Host to ISH (HOST2ISH_MSG15)—Offset 1118h ... 2115
20.97 Inbound Inter Processor Messages 16 Host to ISH (HOST2ISH_MSG16)—Offset 111Ch ... 2116
20.98 Inbound Inter Processor Messages 17 Host to ISH (HOST2ISH_MSG17)—Offset 1120h ... 2116
20.99 Inbound Inter Processor Messages 18 Host to ISH (HOST2ISH_MSG18)—Offset 1124h ... 2116
20.100 Inbound Inter Processor Messages 19 Host to ISH (HOST2ISH_MSG19)—Offset 1128h .. 2117
20.101 Inbound Inter Processor Messages 20 Host to ISH (HOST2ISH_MSG20)—Offset 112Ch .. 2117
20.102 Inbound Inter Processor Messages 21 Host to ISH (HOST2ISH_MSG21)—Offset 1130h .. 2118
20.103 Inbound Inter Processor Messages 22 Host to ISH (HOST2ISH_MSG22)—Offset 1134h .. 2118
20.104 Inbound Inter Processor Messages 23 Host to ISH (HOST2ISH_MSG23)—Offset 1138h .. 2118
20.105 Inbound Inter Processor Messages 24 Host to ISH (HOST2ISH_MSG24)—Offset 113Ch .. 2119
20.106 Inbound Inter Processor Messages 25 Host to ISH (HOST2ISH_MSG25)—Offset 1140h .. 2119
20.107 Inbound Inter Processor Messages 26 Host to ISH (HOST2ISH_MSG26)—Offset 1144h .. 2120
20.108 Inbound Inter Processor Messages 27 Host to ISH (HOST2ISH_MSG27)—Offset 1148h .. 2120
20.109 Inbound Inter Processor Messages 28 Host to ISH (HOST2ISH_MSG28)—Offset 114Ch .. 2120
20.110 Inbound Inter Processor Messages 29 Host to ISH (HOST2ISH_MSG29)—Offset 1150h .. 2121
20.111 Inbound Inter Processor Messages 30 Host to ISH (HOST2ISH_MSG30)—Offset 1154h .. 2121
20.112 Inbound Inter Processor Messages 31 Host to ISH (HOST2ISH_MSG31)—Offset 1158h .. 2122
20.113 Inbound Inter Processor Messages 32 Host to ISH (HOST2ISH_MSG32)—Offset 115Ch .. 2122
20.114 Outbound Inter Processor Messages 1 ISH to SEC (ISH2SEC_MSG1)—Offset 1160h 2122
20.115 Outbound Inter Processor Messages 2 ISH to SEC (ISH2SEC_MSG2)—Offset 1164h 2123
20.116 Outbound Inter Processor Messages 3 ISH to SEC (ISH2SEC_MSG3)—Offset 1168h 2123
20.117 Outbound Inter Processor Messages 4 ISH to SEC (ISH2SEC_MSG4)—Offset 116Ch 2124
20.118 Outbound Inter Processor Messages 5 ISH to SEC (ISH2SEC_MSG5)—Offset 1170h 2124
20.119 Outbound Inter Processor Messages 6 ISH to SEC (ISH2SEC_MSG6)—Offset 1174h 2124
20.120 Outbound Inter Processor Messages 7 ISH to SEC (ISH2SEC_MSG7)—Offset 1178h 2125
20.121 Outbound Inter Processor Messages 8 ISH to SEC (ISH2SEC_MSG8)—Offset 117Ch 2125
20.122 Outbound Inter Processor Messages 9 ISH to SEC (ISH2SEC_MSG9)—Offset 1180h 2126
20.123 Outbound Inter Processor Messages 10 ISH to SEC (ISH2SEC_MSG10)—Offset 1184h ... 2126
20.124 Outbound Inter Processor Messages 11 ISH to SEC (ISH2SEC_MSG11)—Offset 1188h ... 2126

20.125Outbound Inter Processor Messages 12 ISH to SEC (ISH2SEC_MSG12)—Offset 118Ch .. 2127
20.126Outbound Inter Processor Messages 13 ISH to SEC (ISH2SEC_MSG13)—Offset 1190h... 2127
20.127Outbound Inter Processor Messages 14 ISH to SEC (ISH2SEC_MSG14)—Offset 1194h... 2128
20.128Outbound Inter Processor Messages 15 ISH to SEC (ISH2SEC_MSG15)—Offset 1198h... 2128
20.129Outbound Inter Processor Messages 16 ISH to SEC (ISH2SEC_MSG16)—Offset 119Ch .. 2128
20.130Outbound Inter Processor Messages 17 ISH to SEC (ISH2SEC_MSG17)—Offset 11A0h .. 2129
20.131Outbound Inter Processor Messages 18 ISH to SEC (ISH2SEC_MSG18)—Offset 11A4h .. 2129
20.132Outbound Inter Processor Messages 19 ISH to SEC (ISH2SEC_MSG19)—Offset 11A8h .. 2130
20.133Outbound Inter Processor Messages 20 ISH to SEC (ISH2SEC_MSG20)—Offset 11ACh .. 2130
20.134Outbound Inter Processor Messages 21 ISH to SEC (ISH2SEC_MSG21)—Offset 11B0h .. 2130
20.135Outbound Inter Processor Messages 22 ISH to SEC (ISH2SEC_MSG22)—Offset 11B4h .. 2131
20.136Outbound Inter Processor Messages 23 ISH to SEC (ISH2SEC_MSG23)—Offset 11B8h .. 2131
20.137Outbound Inter Processor Messages 24 ISH to SEC (ISH2SEC_MSG24)—Offset 11BCh .. 2132
20.138Outbound Inter Processor Messages 25 ISH to SEC (ISH2SEC_MSG25)—Offset 11C0h .. 2132
20.139Outbound Inter Processor Messages 26 ISH to SEC (ISH2SEC_MSG26)—Offset 11C4h .. 2132
20.140Outbound Inter Processor Messages 27 ISH to SEC (ISH2SEC_MSG27)—Offset 11C8h .. 2133
20.141Outbound Inter Processor Messages 28 ISH to SEC (ISH2SEC_MSG28)—Offset 11CCh .. 2133
20.142Outbound Inter Processor Messages 29 ISH to SEC (ISH2SEC_MSG29)—Offset 11D0h .. 2134
20.143Outbound Inter Processor Messages 30 ISH to SEC (ISH2SEC_MSG30)—Offset 11D4h .. 2134
20.144Outbound Inter Processor Messages 31 ISH to SEC (ISH2SEC_MSG31)—Offset 11D8h .. 2134
20.145Outbound Inter Processor Messages 32 ISH to SEC (ISH2SEC_MSG32)—Offset 11DCh .. 2135
20.146Inbound Inter Processor Messages 1 SEC to ISH (SEC2ISH_MSG1)—Offset 11E0h . 2135
20.147Inbound Inter Processor Messages 2 SEC to ISH (SEC2ISH_MSG2)—Offset 11E4h . 2136
20.148Inbound Inter Processor Messages 3 SEC to ISH (SEC2ISH_MSG3)—Offset 11E8h . 2136
20.149Inbound Inter Processor Messages 4 SEC to ISH (SEC2ISH_MSG4)—Offset 11ECh . 2136
20.150Inbound Inter Processor Messages 5 SEC to ISH (SEC2ISH_MSG5)—Offset 11F0h . 2137
20.151Inbound Inter Processor Messages 6 SEC to ISH (SEC2ISH_MSG6)—Offset 11F4h . 2137
20.152Inbound Inter Processor Messages 7 SEC to ISH (SEC2ISH_MSG7)—Offset 11F8h . 2138
20.153Inbound Inter Processor Messages 8 SEC to ISH (SEC2ISH_MSG8)—Offset 11FCh . 2138
20.154Inbound Inter Processor Messages 9 SEC To ISH (SEC2ISH_MSG9)—Offset 1200h. 2138
20.155Inbound Inter Processor Messages 10 SEC to ISH (SEC2ISH_MSG10)—Offset 1204h..... 2139
20.156Inbound Inter Processor Messages 11 SEC to ISH (SEC2ISH_MSG11)—Offset 1208h..... 2139
20.157Inbound Inter Processor Messages 12 SEC to ISH (SEC2ISH_MSG12)—Offset 120Ch..... 2140

20.158Inbound Inter Processor Messages 13 SEC to ISH (SEC2ISH_MSG13)—Offset 1210h 2140
20.159Inbound Inter Processor Messages 14 SEC to ISH (SEC2ISH_MSG14)—Offset 1214h 2140
20.160Inbound Inter Processor Messages 15 SEC to ISH (SEC2ISH_MSG15)—Offset 1218h 2141
20.161Inbound Inter Processor Messages 16 SEC to ISH (SEC2ISH_MSG16)—Offset 121Ch 2141
20.162Inbound Inter Processor Messages 17 SEC to ISH (SEC2ISH_MSG17)—Offset 1220h 2142
20.163Inbound Inter Processor Messages 18 SEC to ISH (SEC2ISH_MSG18)—Offset 1224h 2142
20.164Inbound Inter Processor Messages 19 SEC to ISH (SEC2ISH_MSG19)—Offset 1228h 2142
20.165Inbound Inter Processor Messages 20 SEC to ISH (SEC2ISH_MSG20)—Offset 122Ch 2143
20.166Inbound Inter Processor Messages 21 SEC to ISH (SEC2ISH_MSG21)—Offset 1230h 2143
20.167Inbound Inter Processor Messages 22 SEC to ISH (SEC2ISH_MSG22)—Offset 1234h 2144
20.168Inbound Inter Processor Messages 23 SEC to ISH (SEC2ISH_MSG23)—Offset 1238h 2144
20.169Inbound Inter Processor Messages 24 SEC to ISH (SEC2ISH_MSG24)—Offset 123Ch 2144
20.170Inbound Inter Processor Messages 25 SEC to ISH (SEC2ISH_MSG25)—Offset 1240h 2145
20.171Inbound Inter Processor Messages 26 SEC to ISH (SEC2ISH_MSG26)—Offset 1244h 2145
20.172Inbound Inter Processor Messages 27 SEC to ISH (SEC2ISH_MSG27)—Offset 1248h 2146
20.173Inbound Inter Processor Messages 28 SEC to ISH (SEC2ISH_MSG28)—Offset 124Ch 2146
20.174Inbound Inter Processor Messages 29 SEC To ISH (SEC2ISH_MSG29)—Offset 1250h..... 2146
20.175Inbound Inter Processor Messages 30 SEC To ISH (SEC2ISH_MSG30)—Offset 1254h..... 2147
20.176Inbound Inter Processor Messages 31 SEC To ISH (SEC2ISH_MSG31)—Offset 1258h..... 2147
20.177Inbound Inter Processor Messages 32 SEC to ISH (SEC2ISH_MSG32)—Offset 125Ch 2148
20.178Outbound Inter Processor Messages 1 ISH to PMC (ISH2PMC_MSG1)—Offset 1260h 2148
20.179Outbound Inter Processor Messages 2 ISH to PMC (ISH2PMC_MSG2)—Offset 1264h 2148
20.180Outbound Inter Processor Messages 3 ISH to PMC (ISH2PMC_MSG3)—Offset 1268h 2149
20.181Outbound Inter Processor Messages 4 ISH to PMC (ISH2PMC_MSG4)—Offset 126Ch 2149
20.182Outbound Inter Processor Messages 5 ISH to PMC (ISH2PMC_MSG5)—Offset 1270h 2150
20.183Outbound Inter Processor Messages 6 ISH to PMC (ISH2PMC_MSG6)—Offset 1274h 2150
20.184Outbound Inter Processor Messages 7 ISH to PMC (ISH2PMC_MSG7)—Offset 1278h 2150
20.185Outbound Inter Processor Messages 8 ISH to PMC (ISH2PMC_MSG8)—Offset 127Ch 2151
20.186Outbound Inter Processor Messages 9 ISH to PMC (ISH2PMC_MSG9)—Offset 1280h .....

2151
20.187Outbound Inter Processor Messages 10 ISH to PMC (ISH2PMC_MSG10)—Offset 1284h .. 2152
20.188Outbound Inter Processor Messages 11 ISH to PMC (ISH2PMC_MSG11)—Offset 1288h .. 2152
20.189Outbound Inter Processor Messages 12 ISH to PMC (ISH2PMC_MSG12)—Offset 128Ch .. 2152
20.190Outbound Inter Processor Messages 13 ISH to PMC (ISH2PMC_MSG13)—Offset 1290h .. 2153
20.191Outbound Inter Processor Messages 14 ISH to PMC (ISH2PMC_MSG14)—Offset 1294h .. 2153
20.192Outbound Inter Processor Messages 15 ISH to PMC (ISH2PMC_MSG15)—Offset 1298h .. 2154
20.193Outbound Inter Processor Messages 16 ISH to PMC (ISH2PMC_MSG16)—Offset 129Ch .. 2154
20.194Outbound Inter Processor Messages 17 ISH to PMC (ISH2PMC_MSG17)—Offset 12A0h .. 2154
20.195Outbound Inter Processor Messages 18 ISH to PMC (ISH2PMC_MSG18)—Offset 12A4h .. 2155
20.196Outbound Inter Processor Messages 19 ISH to PMC (ISH2PMC_MSG19)—Offset 12A8h .. 2155
20.197Outbound Inter Processor Messages 20 ISH To PMC (ISH2PMC_MSG20)—Offset 12ACh . 2156
20.198Outbound Inter Processor Messages 21 ISH to PMC (ISH2PMC_MSG21)—Offset 12B0h .. 2156
20.199Outbound Inter Processor Messages 22 ISH to PMC (ISH2PMC_MSG22)—Offset 12B4h .. 2156
20.200Outbound Inter Processor Messages 23 ISH to PMC (ISH2PMC_MSG23)—Offset 12B8h .. 2157
20.201Outbound Inter Processor Messages 24 ISH to PMC (ISH2PMC_MSG24)—Offset 12BCh .. 2157
20.202Outbound Inter Processor Messages 25 ISH to PMC (ISH2PMC_MSG25)—Offset 12C0h .. 2158
20.203Outbound Inter Processor Messages 26 ISH to PMC (ISH2PMC_MSG26)—Offset 12C4h .. 2158
20.204Outbound Inter Processor Messages 27 ISH to PMC (ISH2PMC_MSG27)—Offset 12C8h .. 2158
20.205Outbound Inter Processor Messages 28 ISH to PMC (ISH2PMC_MSG28)—Offset 12CCh .. 2159
20.206Outbound Inter Processor Messages 29 ISH to PMC (ISH2PMC_MSG29)—Offset 12D0h .. 2159
20.207Outbound Inter Processor Messages 30 ISH to PMC (ISH2PMC_MSG30)—Offset 12D4h .. 2160
20.208Outbound Inter Processor Messages 31 ISH to PMC (ISH2PMC_MSG31)—Offset 12D8h .. 2160
20.209Outbound Inter Processor Messages 32 ISH to PMC (ISH2PMC_MSG32)—Offset 12DCh . 2160
20.210Inbound Inter Processor Messages 1 PMC to ISH (PMC2ISH_MSG1)—Offset 12E0h 2161
20.211Inbound Inter Processor Messages 2 PMC to ISH (PMC2ISH_MSG2)—Offset 12E4h 2161
20.212Inbound Inter Processor Messages 3 PMC to ISH (PMC2ISH_MSG3)—Offset 12E8h 2162
20.213Inbound Inter Processor Messages 4 PMC to ISH (PMC2ISH_MSG4)—Offset 12ECh 2162
20.214Inbound Inter Processor Messages 5 PMC to ISH (PMC2ISH_MSG5)—Offset 12F0h. 2162
20.215Inbound Inter Processor Messages 6 PMC to ISH (PMC2ISH_MSG6)—Offset 12F4h. 2163
20.216Inbound Inter Processor Messages 7 PMC to ISH (PMC2ISH_MSG7)—Offset 12F8h. 2163
20.217Inbound Inter Processor Messages 8 PMC to ISH (PMC2ISH_MSG8)—Offset 12FCh 2164
20.218Inbound Inter Processor Messages 9 PMC to ISH (PMC2ISH_MSG9)—Offset 1300h 2164
20.219Inbound Inter Processor Messages 10 PMC to ISH (PMC2ISH_MSG10)—Offset 1304h ....

2164
20.220Inbound Inter Processor Messages 11 PMC to ISH (PMC2ISH_MSG11)—Offset 1308h..... 2165
20.221Inbound Inter Processor Messages 12 PMC to ISH (PMC2ISH_MSG12)—Offset 130Ch 2165
20.222Inbound Inter Processor Messages 13 PMC to ISH (PMC2ISH_MSG13)—Offset 1310h..... 2166
20.223Inbound Inter Processor Messages 14 PMC to ISH (PMC2ISH_MSG14)—Offset 1314h..... 2166
20.224Inbound Inter Processor Messages 15 PMC to ISH (PMC2ISH_MSG15)—Offset 1318h..... 2166
20.225Inbound Inter Processor Messages 16 PMC to ISH (PMC2ISH_MSG16)—Offset 131Ch 2167
20.226Inbound Inter Processor Messages 17 PMC to ISH (PMC2ISH_MSG17)—Offset 1320h..... 2167
20.227Inbound Inter Processor Messages 18 PMC to ISH (PMC2ISH_MSG18)—Offset 1324h..... 2168
20.228Inbound Inter Processor Messages 19 PMC to ISH (PMC2ISH_MSG19)—Offset 1328h..... 2168
20.229Inbound Inter Processor Messages 20 PMC to ISH (PMC2ISH_MSG20)—Offset 132Ch 2168
20.230Inbound Inter Processor Messages 21 PMC to ISH (PMC2ISH_MSG21)—Offset 1330h..... 2169
20.231Inbound Inter Processor Messages 22 PMC to ISH (PMC2ISH_MSG22)—Offset 1334h..... 2169
20.232Inbound Inter Processor Messages 23 PMC to ISH (PMC2ISH_MSG23)—Offset 1338h..... 2170
20.233Inbound Inter Processor Messages 24 PMC to ISH (PMC2ISH_MSG24)—Offset 133Ch 2170
20.234Inbound Inter Processor Messages 25 PMC to ISH (PMC2ISH_MSG25)—Offset 1340h..... 2170
20.235Inbound Inter Processor Messages 26 PMC to ISH (PMC2ISH_MSG26)—Offset 1344h..... 2171
20.236Inbound Inter Processor Messages 27 PMC to ISH (PMC2ISH_MSG27)—Offset 1348h..... 2171
20.237Inbound Inter Processor Messages 28 PMC to ISH (PMC2ISH_MSG28)—Offset 134Ch 2172
20.238Inbound Inter Processor Messages 29 PMC to ISH (PMC2ISH_MSG29)—Offset 1350h..... 2172
20.239Inbound Inter Processor Messages 30 PMC to ISH (PMC2ISH_MSG30)—Offset 1354h..... 2172
20.240Inbound Inter Processor Messages 31 PMC to ISH (PMC2ISH_MSG31)—Offset 1358h..... 2173
20.241Inbound Inter Processor Messages 32 PMC to ISH (PMC2ISH_MSG32)—Offset 135Ch 2173
20.242Remap 0 (REMAP0)—Offset 1360h 2173
20.243Remap 1 (REMAP1)—Offset 1364h 2174
20.244Remap 2 (REMAP2)—Offset 1368h 2174
20.245Remap 3 (REMAP3)—Offset 136Ch 2174
20.246Remap 4 (REMAP4)—Offset 1370h 2175
20.247Remap 5 (REMAP5)—Offset 1374h 2175
20.248ISH IPC Busy Clear (ISH_IPC_BUSY_CLEAR)—Offset 1378h..... 2175
20.249ISH Fuse (ISH_FUSE)—Offset 137Ch 2176
20.250UMA Base Address LSB (UMA_RANGE_LOWER_0)—Offset 1380h..... 2177
20.251UMA Base Address MSB (UMA_RANGE_LOWER_1)—Offset 1384h 2177
20.252UMA Limit Address LSB (UMA_RANGE_UPPER_0)—Offset 1388h..... 2178
20.253UMA Base Address MSB (UMA_RANGE_UPPER_1)—Offset 138Ch 2178

20.254Outbound Inter Processor Messages 1 ISH to ISP (ISH2ISP_MSG1)—Offset 1390h . 2178
20.255Outbound Inter Processor Messages 2 ISH to ISP (ISH2ISP_MSG2)—Offset 1394h . 2179
20.256Outbound Inter Processor Messages 3 ISH to ISP (ISH2ISP_MSG3)—Offset 1398h . 2179
20.257Outbound Inter Processor Messages 4 ISH to ISP (ISH2ISP_MSG4)—Offset 139Ch . 2180
20.258Outbound Inter Processor Messages 5 ISH to ISP (ISH2ISP_MSG5)—Offset 13A0h . 2180
20.259Outbound Inter Processor Messages 6 ISH to ISP (ISH2ISP_MSG6)—Offset 13A4h . 2180
20.260Outbound Inter Processor Messages 7 ISH to ISP (ISH2ISP_MSG7)—Offset 13A8h . 2181
20.261Outbound Inter Processor Messages 8 ISH to ISP (ISH2ISP_MSG8)—Offset 13ACh 2181
20.262Outbound Inter Processor Messages 9 ISH to ISP (ISH2ISP_MSG9)—Offset 13B0h . 2182
20.263Outbound Inter Processor Messages 10 ISH to ISP (ISH2ISP_MSG10)—Offset 13B4h 2182
20.264Outbound Inter Processor Messages 11 ISH to ISP (ISH2ISP_MSG11)—Offset 13B8h 2183
20.265Outbound Inter Processor Messages 12 ISH to ISP (ISH2ISP_MSG12)—Offset 13BCh 2183
20.266Outbound Inter Processor Messages 13 ISH to ISP (ISH2ISP_MSG13)—Offset 13C0h 2183
20.267Outbound Inter Processor Messages 14 ISH to ISP (ISH2ISP_MSG14)—Offset 13C4h 2184
20.268Outbound Inter Processor Messages 15 ISH to ISP (ISH2ISP_MSG15)—Offset 13C8h 2184
20.269Outbound Inter Processor Messages 16 ISH to ISP (ISH2ISP_MSG16)—Offset 13CCh 2185
20.270Outbound Inter Processor Messages 17 ISH to ISP (ISH2ISP_MSG17)—Offset 13D0h 2185
20.271Outbound Inter Processor Messages 18 ISH to ISP (ISH2ISP_MSG18)—Offset 13D4h 2186
20.272Outbound Inter Processor Messages 19 ISH to ISP (ISH2ISP_MSG19)—Offset 13D8h 2186
20.273Outbound Inter Processor Messages 20 ISH to ISP (ISH2ISP_MSG20)—Offset 13DCh 2186
20.274Outbound Inter Processor Messages 21 ISH To ISP (ISH2ISP_MSG21)—Offset 13E0h 2187
20.275Outbound Inter Processor Messages 22 ISH to ISP (ISH2ISP_MSG22)—Offset 13E4h 2187
20.276Outbound Inter Processor Messages 23 ISH to ISP (ISH2ISP_MSG23)—Offset 13E8h 2188
20.277Outbound Inter Processor Messages 24 ISH to ISP (ISH2ISP_MSG24)—Offset 13ECh 2188
20.278Outbound Inter Processor Messages 25 ISH to ISP (ISH2ISP_MSG25)—Offset 13F0h 2189
20.279Outbound Inter Processor Messages 26 ISH to ISP (ISH2ISP_MSG26)—Offset 13F4h 2189
20.280Outbound Inter Processor Messages 27 ISH to ISP (ISH2ISP_MSG27)—Offset 13F8h 2189
20.281Outbound Inter Processor Messages 28 ISH to ISP (ISH2ISP_MSG28)—Offset 13FCh 2190
20.282Outbound Inter Processor Messages 29 ISH to ISP (ISH2ISP_MSG29)—Offset 1400h 2190
20.283Outbound Inter Processor Messages 30 ISH to ISP (ISH2ISP_MSG30)—Offset 1404h 2191
20.284Outbound Inter Processor Messages 31 ISH to ISP (ISH2ISP_MSG31)—Offset 1408h 2191
20.285Outbound Inter Processor Messages 32 ISH to ISP (ISH2ISP_MSG32)—Offset 140Ch 2192
20.286Inbound Inter Processor Messages 1 ISP to ISH (ISP2ISH_MSG1)—Offset 1410h ... 2192
20.287Inbound Inter Processor Messages 2 ISP to ISH (ISP2ISH_MSG2)—Offset 1414h ... 2192

20.288Inbound Inter Processor Messages 3 ISP to ISH (ISP2ISH_MSG3)—Offset 1418h ...2193
20.289Inbound Inter Processor Messages 4 ISP to ISH (ISP2ISH_MSG4)—Offset 141Ch ...2193
20.290Inbound Inter Processor Messages 5 ISP to ISH (ISP2ISH_MSG5)—Offset 1420h ...2194
20.291Inbound Inter Processor Messages 6 ISP to ISH (ISP2ISH_MSG6)—Offset 1424h ...2194
20.292Inbound Inter Processor Messages 7 ISP to ISH (ISP2ISH_MSG7)—Offset 1428h ...2195
20.293Inbound Inter Processor Messages 8 ISP to ISH (ISP2ISH_MSG8)—Offset 142Ch ...2195
20.294Inbound Inter Processor Messages 9 ISP to ISH (ISP2ISH_MSG9)—Offset 1430h ...2195
20.295Inbound Inter Processor Messages 10 ISP to ISH (ISP2ISH_MSG10)—Offset 1434h 2196
20.296Inbound Inter Processor Messages 11 ISP to ISH (ISP2ISH_MSG11)—Offset 1438h 2196
20.297Inbound Inter Processor Messages 12 ISP to ISH (ISP2ISH_MSG12)—Offset 143Ch 2197
20.298Inbound Inter Processor Messages 13 ISP to ISH (ISP2ISH_MSG13)—Offset 1440h 2197
20.299Inbound Inter Processor Messages 14 ISP to ISH (ISP2ISH_MSG14)—Offset 1444h 2198
20.300Inbound Inter Processor Messages 15 ISP to ISH (ISP2ISH_MSG15)—Offset 1448h 2198
20.301Inbound Inter Processor Messages 16 ISP to ISH (ISP2ISH_MSG16)—Offset 144Ch 2198
20.302Inbound Inter Processor Messages 17 ISP to ISH (ISP2ISH_MSG17)—Offset 1450h 2199
20.303Inbound Inter Processor Messages 18 ISP to ISH (ISP2ISH_MSG18)—Offset 1454h 2199
20.304Inbound Inter Processor Messages 19 ISP to ISH (ISP2ISH_MSG19)—Offset 1458h 2200
20.305Inbound Inter Processor Messages 20 ISP to ISH (ISP2ISH_MSG20)—Offset 145Ch 2200
20.306Inbound Inter Processor Messages 21 ISP to ISH (ISP2ISH_MSG21)—Offset 1460h 2201
20.307Inbound Inter Processor Messages 22 ISP to ISH (ISP2ISH_MSG22)—Offset 1464h 2201
20.308Inbound Inter Processor Messages 23 ISP to ISH (ISP2ISH_MSG23)—Offset 1468h 2201
20.309Inbound Inter Processor Messages 24 ISP to ISH (ISP2ISH_MSG24)—Offset 146Ch 2202
20.310Inbound Inter Processor Messages 25 ISP to ISH (ISP2ISH_MSG25)—Offset 1470h 2202
20.311Inbound Inter Processor Messages 26 ISP to ISH (ISP2ISH_MSG26)—Offset 1474h 2203
20.312Inbound Inter Processor Messages 27 ISP to ISH (ISP2ISH_MSG27)—Offset 1478h 2203
20.313Inbound Inter Processor Messages 28 ISP to ISH (ISP2ISH_MSG28)—Offset 147Ch 2204
20.314Inbound Inter Processor Messages 29 ISP to ISH (ISP2ISH_MSG29)—Offset 1480h 2204
20.315Inbound Inter Processor Messages 30 ISP to ISH (ISP2ISH_MSG30)—Offset 1484h 2204
20.316Inbound Inter Processor Messages 31 ISP to ISH (ISP2ISH_MSG31)—Offset 1488h 2205
20.317Inbound Inter Processor Messages 32 ISP to ISH (ISP2ISH_MSG32)—Offset 148Ch 2205
20.318Outbound Inter Processor Messages 1 ISH to Audio (ISH2AUDIO_MSG1)—Offset 1490h. 2206
20.319Outbound Inter Processor Messages 2 ISH to Audio (ISH2AUDIO_MSG2)—Offset 1494h. 2206
20.320Outbound Inter Processor Messages 3 ISH to Audio (ISH2AUDIO_MSG3)—Offset 1498h. 2207
20.321Outbound Inter Processor Messages 4 ISH to Audio (ISH2AUDIO_MSG4)—Offset 149Ch 2207
20.322Outbound Inter Processor Messages 5 ISH to Audio (ISH2AUDIO_MSG5)—Offset 14A0h. 2207
20.323Outbound Inter Processor Messages 6 ISH to Audio (ISH2AUDIO_MSG6)—Offset 14A4h. 2208
20.324Outbound Inter Processor Messages 7 ISH to Audio (ISH2AUDIO_MSG7)—Offset 14A8h. 2208
20.325Outbound Inter Processor Messages 8 ISH to Audio (ISH2AUDIO_MSG8)—Offset 14ACh 2209
20.326Outbound Inter Processor Messages 9 ISH to Audio (ISH2AUDIO_MSG9)—Offset 14B0h. 2209
20.327Outbound Inter Processor Messages 10 ISH to Audio (ISH2AUDIO_MSG10)—Offset 14B4h 2210
20.328Outbound Inter Processor Messages 11 ISH to Audio (ISH2AUDIO_MSG11)—Offset 14B8h 2210
20.329Outbound Inter Processor Messages 12 ISH to Audio (ISH2AUDIO_MSG12)—Offset 14BCh 2210
20.330Outbound Inter Processor Messages 13 ISH to Audio (ISH2AUDIO_MSG13)—Offset 14C0h 2211

20.331Outbound Inter Processor Messages 14 ISH to Audio (ISH2AUDIO_MSG14)—Offset 14C4h 2211
20.332Outbound Inter Processor Messages 15 ISH to Audio (ISH2AUDIO_MSG15)—Offset 14C8h 2212
20.333Outbound Inter Processor Messages 16 ISH to Audio (ISH2AUDIO_MSG16)—Offset 14CCh 2212
20.334Outbound Inter Processor Messages 17 ISH to Audio (ISH2AUDIO_MSG17)—Offset 14D0h 2213
20.335Outbound Inter Processor Messages 18 ISH to Audio (ISH2AUDIO_MSG18)—Offset 14D4h 2213
20.336Outbound Inter Processor Messages 19 ISH to Audio (ISH2AUDIO_MSG19)—Offset 14D8h 2213
20.337Outbound Inter Processor Messages 20 ISH to Audio (ISH2AUDIO_MSG20)—Offset 14DCh 2214
20.338Outbound Inter Processor Messages 21 ISH to Audio (ISH2AUDIO_MSG21)—Offset 14E0h 2214
20.339Outbound Inter Processor Messages 22 ISH to Audio (ISH2AUDIO_MSG22)—Offset 14E4h 2215
20.340Outbound Inter Processor Messages 23 ISH to Audio (ISH2AUDIO_MSG23)—Offset 14E8h 2215
20.341Outbound Inter Processor Messages 24 ISH to Audio (ISH2AUDIO_MSG24)—Offset 14ECh 2216
20.342Outbound Inter Processor Messages 25 ISH to Audio (ISH2AUDIO_MSG25)—Offset 14F0h 2216
20.343Outbound Inter Processor Messages 26 ISH to Audio (ISH2AUDIO_MSG26)—Offset 14F4h 2216
20.344Outbound Inter Processor Messages 27 ISH to Audio (ISH2AUDIO_MSG27)—Offset 14F8h 2217
20.345Outbound Inter Processor Messages 28 ISH to Audio (ISH2AUDIO_MSG28)—Offset 14FCh 2217
20.346Outbound Inter Processor Messages 29 ISH to Audio (ISH2AUDIO_MSG29)—Offset 1500h 2218
20.347Outbound Inter Processor Messages 30 ISH to Audio (ISH2AUDIO_MSG30)—Offset 1504h 2218
20.348Outbound Inter Processor Messages 31 ISH to Audio (ISH2AUDIO_MSG31)—Offset 1508h 2219
20.349Outbound Inter Processor Messages 32 ISH to Audio (ISH2AUDIO_MSG32)—Offset 150Ch 2219
20.350Inbound Inter Processor Messages 1 Audio to ISH (AUDIO2ISH_MSG1)—Offset 1510h .. 2219
20.351Inbound Inter Processor Messages 2 Audio to ISH (AUDIO2ISH_MSG2)—Offset 1514h .. 2220
20.352Inbound Inter Processor Messages 3 Audio to ISH (AUDIO2ISH_MSG3)—Offset 1518h .. 2220
20.353Inbound Inter Processor Messages 4 Audio to ISH (AUDIO2ISH_MSG4)—Offset 151Ch .. 2221
20.354Inbound Inter Processor Messages 5 Audio to ISH (AUDIO2ISH_MSG5)—Offset 1520h .. 2221
20.355Inbound Inter Processor Messages 6 Audio to ISH (AUDIO2ISH_MSG6)—Offset 1524h .. 2222
20.356Inbound Inter Processor Messages 7 Audio to ISH (AUDIO2ISH_MSG7)—Offset 1528h .. 2222
20.357Inbound Inter Processor Messages 8 Audio to ISH (AUDIO2ISH_MSG8)—Offset 152Ch .. 2222
20.358Inbound Inter Processor Messages 9 Audio to ISH (AUDIO2ISH_MSG9)—Offset 1530h .. 2223
20.359Inbound Inter Processor Messages 10 Audio to ISH (AUDIO2ISH_MSG10)—Offset 1534h

2223
20.360Inbound Inter Processor Messages 11 Audio to ISH (AUDIO2ISH_MSG11)—Offset 1538h 2224
20.361Inbound Inter Processor Messages 12 Audio to ISH (AUDIO2ISH_MSG12)—Offset 153Ch 2224
20.362Inbound Inter Processor Messages 13 Audio to ISH (AUDIO2ISH_MSG13)—Offset 1540h 2225
20.363Inbound Inter Processor Messages 14 Audio to ISH (AUDIO2ISH_MSG14)—Offset 1544h 2225
20.364Inbound Inter Processor Messages 15 Audio to ISH (AUDIO2ISH_MSG15)—Offset 1548h 2225
20.365Inbound Inter Processor Messages 16 Audio to ISH (AUDIO2ISH_MSG16)—Offset 154Ch 2226
20.366Inbound Inter Processor Messages 17 Audio to ISH (AUDIO2ISH_MSG17)—Offset 1550h 2226
20.367Inbound Inter Processor Messages 18 Audio to ISH (AUDIO2ISH_MSG18)—Offset 1554h 2227
20.368Inbound Inter Processor Messages 19 Audio to ISH (AUDIO2ISH_MSG19)—Offset 1558h 2227
20.369Inbound Inter Processor Messages 20 Audio to ISH (AUDIO2ISH_MSG20)—Offset 155Ch 2228
20.370Inbound Inter Processor Messages 21 Audio to ISH (AUDIO2ISH_MSG21)—Offset 1560h 2228
20.371Inbound Inter Processor Messages 22 Audio to ISH (AUDIO2ISH_MSG22)—Offset 1564h 2228
20.372Inbound Inter Processor Messages 23 Audio to ISH (AUDIO2ISH_MSG23)—Offset 1568h 2229
20.373Inbound Inter Processor Messages 24 Audio to ISH (AUDIO2ISH_MSG24)—Offset 156Ch 2229
20.374Inbound Inter Processor Messages 25 Audio to ISH (AUDIO2ISH_MSG25)—Offset 1570h 2230
20.375Inbound Inter Processor Messages 26 Audio to ISH (AUDIO2ISH_MSG26)—Offset 1574h 2230
20.376Inbound Inter Processor Messages 27 Audio to ISH (AUDIO2ISH_MSG27)—Offset 1578h 2231
20.377Inbound Inter Processor Messages 28 Audio to ISH (AUDIO2ISH_MSG28)—Offset 157Ch 2231
20.378Inbound Inter Processor Messages 29 Audio to ISH (AUDIO2ISH_MSG29)—Offset 1580h 2231
20.379Inbound Inter Processor Messages 30 Audio to ISH (AUDIO2ISH_MSG30)—Offset 1584h 2232
20.380Inbound Inter Processor Messages 31 Audio to ISH (AUDIO2ISH_MSG31)—Offset 1588h 2232
20.381Inbound Inter Processor Messages 32 Audio to ISH (AUDIO2ISH_MSG32)—Offset 158Ch 2233
20.382Outbound Inter Processor Messages 1 ISH to Gfx (ISH2GFX_MSG1)—Offset 1590h 2233
20.383Outbound Inter Processor Messages 2 ISH to Gfx (ISH2GFX_MSG2)—Offset 1594h 2234
20.384Outbound Inter Processor Messages 3 ISH to Gfx (ISH2GFX_MSG3)—Offset 1598h 2234
20.385Outbound Inter Processor Messages 4 ISH to Gfx (ISH2GFX_MSG4)—Offset 159Ch 2234
20.386Outbound Inter Processor Messages 5 ISH to Gfx (ISH2GFX_MSG5)—Offset 15A0h 2235
20.387Outbound Inter Processor Messages 6 ISH to Gfx (ISH2GFX_MSG6)—Offset 15A4h 2235
20.388Outbound Inter Processor Messages 7 ISH to Gfx (ISH2GFX_MSG7)—Offset 15A8h 2236
20.389Outbound Inter Processor Messages 8 ISH to Gfx (ISH2GFX_MSG8)—Offset 15ACh 2236
20.390Outbound Inter Processor Messages 9 ISH to Gfx (ISH2GFX_MSG9)—Offset 15B0h 2237
20.391Outbound Inter Processor Messages 10 ISH to Gfx (ISH2GFX_MSG10)—Offset 15B4h.... 2237
20.392Outbound Inter Processor Messages 11 ISH to Gfx (ISH2GFX_MSG11)—Offset 15B8h....

2237
20.393Outbound Inter Processor Messages 12 ISH to Gfx (ISH2GFX_MSG12)—Offset 15BCh ... 2238
20.394Outbound Inter Processor Messages 13 ISH to Gfx (ISH2GFX_MSG13)—Offset 15C0h ... 2238
20.395Outbound Inter Processor Messages 14 ISH to Gfx (ISH2GFX_MSG14)—Offset 15C4h ... 2239
20.396Outbound Inter Processor Messages 15 ISH to Gfx (ISH2GFX_MSG15)—Offset 15C8h ... 2239
20.397Outbound Inter Processor Messages 16 ISH to Gfx (ISH2GFX_MSG16)—Offset 15CCh ... 2240
20.398Outbound Inter Processor Messages 17 ISH to Gfx (ISH2GFX_MSG17)—Offset 15D0h ... 2240
20.399Outbound Inter Processor Messages 18 ISH to Gfx (ISH2GFX_MSG18)—Offset 15D4h ... 2240
20.400Outbound Inter Processor Messages 19 ISH to Gfx (ISH2GFX_MSG19)—Offset 15D8h ... 2241
20.401Outbound Inter Processor Messages 20 ISH to Gfx (ISH2GFX_MSG20)—Offset 15DCh ... 2241
20.402Outbound Inter Processor Messages 21 ISH to Gfx (ISH2GFX_MSG21)—Offset 15E0h ... 2242
20.403Outbound Inter Processor Messages 22 ISH to Gfx (ISH2GFX_MSG22)—Offset 15E4h ... 2242
20.404Outbound Inter Processor Messages 23 ISH to Gfx (ISH2GFX_MSG23)—Offset 15E8h ... 2243
20.405Outbound Inter Processor Messages 24 ISH to Gfx (ISH2GFX_MSG24)—Offset 15ECh ... 2243
20.406Outbound Inter Processor Messages 25 ISH to Gfx (ISH2GFX_MSG25)—Offset 15F0h.... 2243
20.407Outbound Inter Processor Messages 26 ISH to Gfx (ISH2GFX_MSG26)—Offset 15F4h.... 2244
20.408Outbound Inter Processor Messages 27 ISH to Gfx (ISH2GFX_MSG27)—Offset 15F8h.... 2244
20.409Outbound Inter Processor Messages 28 ISH to Gfx (ISH2GFX_MSG28)—Offset 15FCh ... 2245
20.410Outbound Inter Processor Messages 29 ISH to Gfx (ISH2GFX_MSG29)—Offset 1600h ... 2245
20.411Outbound Inter Processor Messages 30 ISH to Gfx (ISH2GFX_MSG30)—Offset 1604h ... 2246
20.412Outbound Inter Processor Messages 31 ISH to Gfx (ISH2GFX_MSG31)—Offset 1608h ... 2246
20.413Outbound Inter Processor Messages 32 ISH to Gfx (ISH2GFX_MSG32)—Offset 160Ch ... 2246
20.414Inbound Inter Processor Messages 1 Gfx to ISH (GFX2ISH_MSG1)—Offset 1610h .. 2247
20.415Inbound Inter Processor Messages 2 Gfx to ISH (GFX2ISH_MSG2)—Offset 1614h .. 2247
20.416Inbound Inter Processor Messages 3 Gfx to ISH (GFX2ISH_MSG3)—Offset 1618h .. 2248
20.417Inbound Inter Processor Messages 4 Gfx to ISH (GFX2ISH_MSG4)—Offset 161Ch .. 2248
20.418Inbound Inter Processor Messages 5 Gfx to ISH (GFX2ISH_MSG5)—Offset 1620h .. 2249
20.419Inbound Inter Processor Messages 6 Gfx to ISH (GFX2ISH_MSG6)—Offset 1624h .. 2249
20.420Inbound Inter Processor Messages 7 Gfx to ISH (GFX2ISH_MSG7)—Offset 1628h .. 2249
20.421Inbound Inter Processor Messages 8 Gfx to ISH (GFX2ISH_MSG8)—Offset 162Ch .. 2250
20.422Inbound Inter Processor Messages 9 Gfx to ISH (GFX2ISH_MSG9)—Offset 1630h .. 2250
20.423Inbound Inter Processor Messages 10 Gfx to ISH (GFX2ISH_MSG10)—Offset 1634h 2251
20.424Inbound Inter Processor Messages 11 Gfx to ISH (GFX2ISH_MSG11)—Offset 1638h 2251
20.425Inbound Inter Processor Messages 12 Gfx to ISH (GFX2ISH_MSG12)—Offset 163Ch .....

2252
20.426Inbound Inter Processor Messages 13 Gfx to ISH (GFX2ISH_MSG13)—Offset 1640h 2252
20.427Inbound Inter Processor Messages 14 Gfx to ISH (GFX2ISH_MSG14)—Offset 1644h 2252
20.428Inbound Inter Processor Messages 15 Gfx to ISH (GFX2ISH_MSG15)—Offset 1648h 2253
20.429Inbound Inter Processor Messages 16 Gfx to ISH (GFX2ISH_MSG16)—Offset 164Ch 2253
20.430Inbound Inter Processor Messages 17 Gfx to ISH (GFX2ISH_MSG17)—Offset 1650h 2254
20.431Inbound Inter Processor Messages 18 Gfx to ISH (GFX2ISH_MSG18)—Offset 1654h 2254
20.432Inbound Inter Processor Messages 19 Gfx to ISH (GFX2ISH_MSG19)—Offset 1658h 2255
20.433Inbound Inter Processor Messages 20 Gfx to ISH (GFX2ISH_MSG20)—Offset 165Ch 2255
20.434Inbound Inter Processor Messages 21 Gfx to ISH (GFX2ISH_MSG21)—Offset 1660h 2255
20.435Inbound Inter Processor Messages 22 Gfx to ISH (GFX2ISH_MSG22)—Offset 1664h 2256
20.436Inbound Inter Processor Messages 23 Gfx to ISH (GFX2ISH_MSG23)—Offset 1668h 2256
20.437Inbound Inter Processor Messages 24 Gfx to ISH (GFX2ISH_MSG24)—Offset 166Ch 2257
20.438Inbound Inter Processor Messages 25 Gfx to ISH (GFX2ISH_MSG25)—Offset 1670h 2257
20.439Inbound Inter Processor Messages 26 Gfx to ISH (GFX2ISH_MSG26)—Offset 1674h 2258
20.440Inbound Inter Processor Messages 27 Gfx to ISH (GFX2ISH_MSG27)—Offset 1678h 2258
20.441Inbound Inter Processor Messages 28 Gfx to ISH (GFX2ISH_MSG28)—Offset 167Ch 2258
20.442Inbound Inter Processor Messages 29 Gfx to ISH (GFX2ISH_MSG29)—Offset 1680h 2259
20.443Inbound Inter Processor Messages 30 Gfx to ISH (GFX2ISH_MSG30)—Offset 1684h 2259
20.444Inbound Inter Processor Messages 31 Gfx to ISH (GFX2ISH_MSG31)—Offset 1688h 2260
20.445Inbound Inter Processor Messages 32 Gfx to ISH (GFX2ISH_MSG32)—Offset 168Ch 2260
20.446Inbound Doorbell ISP to ISH (ISP2ISH_DOORBELL)—Offset 1690h 2261
20.447Outbound Doorbell ISH to ISP (ISH2ISP_DOORBELL)—Offset 1694h 2261
20.448Inbound Doorbell Audio to ISH (AUDIO2ISH_DOORBELL)—Offset 1698h 2261
20.449Outbound Doorbell ISH to Audio (ISH2AUDIO_DOORBELL)—Offset 169Ch 2262
20.450Inbound Doorbell GFX to ISH (GFX2ISH_DOORBELL)—Offset 16A0h 2262
20.451Outbound Doorbell ISH to GFx (ISH2GFX_DOORBELL)—Offset 16A4h 2263
20.452CSME to ISH Control and Status (CSME2ISH_CTRL_STATUS)—Offset 16A8h.....2263
20.453ISH IPC D0I3 Control (IPC_d0i3C_reg)—Offset 16D0h 2264
20.454PMC to ISH Control and Status (ipc_pmc2ish_csr_reg)—Offset 16D4h.....2264
20.455ISP to ISH Control and Status (ipc_isp2ish_csr_reg)—Offset 16D8h.....2265
20.456Audio to ISH Control and Status (ipc_audio2ish_csr_reg)—Offset 16DCh.....2265
20.457Peripheral Interrupt Status (PISR)—Offset 0h 2265
20.458Peripheral Interrupt Mask (PIMR)—Offset 4h 2267
20.459Host Peripheral Interrupt Mask (HOST_PIMR)—Offset 8h.....2269
20.460HOST Peripheral Interrupt Status (HOST_PISR)—Offset Ch.....2269

20.461Reserved (RESERVED2)—Offset 10h 2270
20.462Reserved (RESERVED3)—Offset 14h 2270
20.463Reserved (RESERVED4)—Offset 18h 2270
20.464Reserved (RESERVED5)—Offset 1Ch 2271
20.465Reserved (RESERVED6)—Offset 20h 2271
20.466Reserved (RESERVED7)—Offset 24h 2271
20.467Reserved (RESERVED8)—Offset 28h 2272
20.468Reserved (RESERVED9)—Offset 2Ch 2272
20.469Reserved (RESERVED10)—Offset 30h 2272
20.470ISH Host Firmware Status (ISH_HOST_FWSTS)—Offset 34h 2273
20.471Host Communication (HOST_COMM)—Offset 38h..... 2273
20.472ISH SEC Firmware Status (ISH_SEC_FWSTS)—Offset 3Ch 2273
20.473SEC Communication (SEC_COMM)—Offset 40h..... 2274
20.474ISH Reset (ISH_RST)—Offset 44h 2274
20.475Inbound Doorbell HOST to ISH (HOST2ISH_DOORBELL)—Offset 48h..... 2274
20.476Inbound Doorbell PMC to ISH (PMC2ISH_DOORBELL)—Offset 4Ch 2275
20.477Inbound Doorbell SEC to ISH (SEC2ISH_DOORBELL)—Offset 50h 2275
20.478Outbound Doorbell ISH to Host (ISH2HOST_DOORBELL)—Offset 54h 2276
20.479Outbound Doorbell ISH to PMC (ISH2PMC_DOORBELL)—Offset 58h 2276
20.480Outbound Doorbell ISH to SEC (ISH2SEC_DOORBELL)—Offset 5Ch 2276
20.481Outbound Inter-processor Messages 1 ISH to Host (ISH2HOST_MSG1)—Offset 60h 2277
20.482Outbound Inter-processor Messages 2 ISH to Host (ISH2HOST_MSG2)—Offset 64h 2277
20.483Outbound Inter-processor Messages 3 ISH to Host (ISH2HOST_MSG3)—Offset 68h 2278
20.484Outbound Inter-processor Messages 4 ISH to Host (ISH2HOST_MSG4)—Offset 6Ch 2278
20.485Outbound Inter-processor Messages 5 ISH to Host (ISH2HOST_MSG5)—Offset 70h 2278
20.486Outbound Inter-processor Messages 6 ISH to Host (ISH2HOST_MSG6)—Offset 74h 2279
20.487Outbound Inter-processor Messages 7 ISH to Host (ISH2HOST_MSG7)—Offset 78h 2279
20.488Outbound Inter-processor Messages 8 ISH to Host (ISH2HOST_MSG8)—Offset 7Ch 2280
20.489Outbound Inter-processor Messages 9 ISH to Host (ISH2HOST_MSG9)—Offset 80h 2280
20.490Outbound Inter-processor Messages 10 ISH to Host (ISH2HOST_MSG10)—Offset 84h... 2280
20.491Outbound Inter-processor Messages 11 ISH to Host (ISH2HOST_MSG11)—Offset 88h... 2281
20.492Outbound Inter-processor Messages 12 ISH to Host (ISH2HOST_MSG12)—Offset 8Ch... 2281
20.493Outbound Inter-processor Messages 13 ISH to Host (ISH2HOST_MSG13)—Offset 90h... 2282
20.494Outbound Inter-processor Messages 14 ISH to Host (ISH2HOST_MSG14)—Offset 94h... 2282
20.495Outbound Inter-processor Messages 15 ISH to Host (ISH2HOST_MSG15)—Offset 98h... 2282
20.496Outbound Inter-processor Messages 16 ISH to Host (ISH2HOST_MSG16)—Offset 9Ch... 2283
20.497Outbound Inter-processor Messages 17 ISH to Host (ISH2HOST_MSG17)—Offset A0h... 2283
20.498Outbound Inter-processor Messages 18 ISH to Host (ISH2HOST_MSG18)—Offset A4h... 2284
20.499Outbound Inter-processor Messages 19 ISH to Host (ISH2HOST_MSG19)—Offset A8h... 2284
20.500Outbound Inter-processor Messages 20 ISH to Host (ISH2HOST_MSG20)—Offset ACh .. 2284
20.501Outbound Inter-processor Messages 21 ISH to Host (ISH2HOST_MSG21)—Offset B0h... 2285
20.502Outbound Inter-processor Messages 22 ISH to Host (ISH2HOST_MSG22)—Offset B4h... 2285
20.503Outbound Inter-processor Messages 23 ISH to Host (ISH2HOST_MSG23)—Offset B8h...

2286
20.504Outbound Inter-processor Messages 24 ISH to Host (ISH2HOST_MSG24)—Offset BCh ... 2286
20.505Outbound Inter-processor Messages 25 ISH to Host (ISH2HOST_MSG25)—Offset C0h ... 2286
20.506Outbound Inter-processor Messages 26 ISH to Host (ISH2HOST_MSG26)—Offset C4h ... 2287
20.507Outbound Inter-processor Messages 27 ISH to Host (ISH2HOST_MSG27)—Offset C8h ... 2287
20.508Outbound Inter-processor Messages 28 ISH to Host (ISH2HOST_MSG28)—Offset CCh ... 2288
20.509Outbound Inter-processor Messages 29 ISH to Host (ISH2HOST_MSG29)—Offset D0h ... 2288
20.510Outbound Inter-processor Messages 30 ISH to Host (ISH2HOST_MSG30)—Offset D4h ... 2288
20.511Outbound Inter-processor Messages 31 ISH to Host (ISH2HOST_MSG31)—Offset D8h ... 2289
20.512Outbound Inter-processor Messages 32 ISH to Host (ISH2HOST_MSG32)—Offset DCh ... 2289
20.513Inbound Inter-processor Messages 1 Host to ISH (HOST2ISH_MSG1)—Offset E0h ..2290
20.514Inbound Inter-processor Messages 2 Host to ISH (HOST2ISH_MSG2)—Offset E4h ..2290
20.515Inbound Inter-processor Messages 3 Host to ISH (HOST2ISH_MSG3)—Offset E8h ..2290
20.516Inbound Inter-processor Messages 4 Host to ISH (HOST2ISH_MSG4)—Offset ECh ..2291
20.517Inbound Inter-processor Messages 5 Host to ISH (HOST2ISH_MSG5)—Offset F0h ..2291
20.518Inbound Inter-processor Messages 6 Host to ISH (HOST2ISH_MSG6)—Offset F4h ..2292
20.519Inbound Inter-processor Messages 7 Host to ISH (HOST2ISH_MSG7)—Offset F8h ..2292
20.520Inbound Inter-processor Messages 8 Host to ISH (HOST2ISH_MSG8)—Offset FCh ..2292
20.521Inbound Inter-processor Messages 9 Host to ISH (HOST2ISH_MSG9)—Offset 100h 2293
20.522Inbound Inter-processor Messages 10 Host to ISH (HOST2ISH_MSG10)—Offset 104h.... 2293
20.523Inbound Inter-processor Messages 11 Host to ISH (HOST2ISH_MSG11)—Offset 108h.... 2294
20.524Inbound Inter-processor Messages 12 Host to ISH (HOST2ISH_MSG12)—Offset 10Ch.... 2294
20.525Inbound Inter-processor Messages 13 Host to ISH (HOST2ISH_MSG13)—Offset 110h.... 2294
20.526Inbound Inter-processor Messages 14 Host to ISH (HOST2ISH_MSG14)—Offset 114h.... 2295
20.527Inbound Inter-processor Messages 15 Host to ISH (HOST2ISH_MSG15)—Offset 118h.... 2295
20.528Inbound Inter-processor Messages 16 Host to ISH (HOST2ISH_MSG16)—Offset 11Ch.... 2296
20.529Inbound Inter-processor Messages 17 Host to ISH (HOST2ISH_MSG17)—Offset 120h.... 2296
20.530Inbound Inter-processor Messages 18 Host to ISH (HOST2ISH_MSG18)—Offset 124h.... 2296
20.531Inbound Inter-processor Messages 19 Host to ISH (HOST2ISH_MSG19)—Offset 128h.... 2297
20.532Inbound Inter-processor Messages 20 Host to ISH (HOST2ISH_MSG20)—Offset 12Ch.... 2297
20.533Inbound Inter-processor Messages 21 Host to ISH (HOST2ISH_MSG21)—Offset 130h.... 2298
20.534Inbound Inter-processor Messages 22 Host to ISH (HOST2ISH_MSG22)—Offset 134h.... 2298
20.535Inbound Inter-processor Messages 23 Host to ISH (HOST2ISH_MSG23)—Offset 138h.... 2298
20.536Inbound Inter-processor Messages 24 Host to ISH (HOST2ISH_MSG24)—Offset 13Ch....

2299
20.537Inbound Inter-processor Messages 25 Host to ISH (HOST2ISH_MSG25)—Offset 140h ... 2299
20.538Inbound Inter-processor Messages 26 Host to ISH (HOST2ISH_MSG26)—Offset 144h ... 2300
20.539Inbound Inter-processor Messages 27 Host to ISH (HOST2ISH_MSG27)—Offset 148h ... 2300
20.540Inbound Inter-processor Messages 28 Host to ISH (HOST2ISH_MSG28)—Offset 14Ch ... 2300
20.541Inbound Inter-processor Messages 29 Host to ISH (HOST2ISH_MSG29)—Offset 150h ... 2301
20.542Inbound Inter-processor Messages 30 Host to ISH (HOST2ISH_MSG30)—Offset 154h ... 2301
20.543Inbound Inter-processor Messages 31 Host to ISH (HOST2ISH_MSG31)—Offset 158h ... 2302
20.544Inbound Inter-processor Messages 32 Host to ISH (HOST2ISH_MSG32)—Offset 15Ch ... 2302
20.545Outbound Inter-processor Messages 1 ISH to SEC (ISH2SEC_MSG1)—Offset 160h. 2302
20.546Outbound Inter-processor Messages 2 ISH to SEC (ISH2SEC_MSG2)—Offset 164h. 2303
20.547Outbound Inter-processor Messages 3 ISH to SEC (ISH2SEC_MSG3)—Offset 168h. 2303
20.548Outbound Inter-processor Messages 4 ISH to SEC (ISH2SEC_MSG4)—Offset 16Ch 2304
20.549Outbound Inter-processor Messages 5 ISH to SEC (ISH2SEC_MSG5)—Offset 170h. 2304
20.550Outbound Inter-processor Messages 6 ISH to SEC (ISH2SEC_MSG6)—Offset 174h. 2304
20.551Outbound Inter-processor Messages 7 ISH to SEC (ISH2SEC_MSG7)—Offset 178h. 2305
20.552Outbound Inter-processor Messages 8 ISH to SEC (ISH2SEC_MSG8)—Offset 17Ch 2305
20.553Outbound Inter-processor Messages 9 ISH to SEC (ISH2SEC_MSG9)—Offset 180h. 2306
20.554Outbound Inter-processor Messages 10 ISH to SEC (ISH2SEC_MSG10)—Offset 184h 2306
20.555Outbound Inter-processor Messages 11 ISH to SEC (ISH2SEC_MSG11)—Offset 188h 2306
20.556Outbound Inter-processor Messages 12 ISH to SEC (ISH2SEC_MSG12)—Offset 18Ch 2307
20.557Outbound Inter-processor Messages 13 ISH to SEC (ISH2SEC_MSG13)—Offset 190h 2307
20.558Outbound Inter-processor Messages 14 ISH to SEC (ISH2SEC_MSG14)—Offset 194h 2308
20.559Outbound Inter-processor Messages 15 ISH to SEC (ISH2SEC_MSG15)—Offset 198h 2308
20.560Outbound Inter-processor Messages 16 ISH to SEC (ISH2SEC_MSG16)—Offset 19Ch 2308
20.561Outbound Inter-processor Messages 17 ISH to SEC (ISH2SEC_MSG17)—Offset 1A0h 2309
20.562Outbound Inter-processor Messages 18 ISH to SEC (ISH2SEC_MSG18)—Offset 1A4h 2309
20.563Outbound Inter-processor Messages 19 ISH to SEC (ISH2SEC_MSG19)—Offset 1A8h 2310
20.564Outbound Inter-processor Messages 20 ISH to SEC (ISH2SEC_MSG20)—Offset 1ACh 2310
20.565Outbound Inter-processor Messages 21 ISH to SEC (ISH2SEC_MSG21)—Offset 1B0h 2310
20.566Outbound Inter-processor Messages 22 ISH to SEC (ISH2SEC_MSG22)—Offset 1B4h 2311
20.567Outbound Inter-processor Messages 23 ISH to SEC (ISH2SEC_MSG23)—Offset 1B8h 2311
20.568Outbound Inter-processor Messages 24 ISH to SEC (ISH2SEC_MSG24)—Offset 1BCh 2312
20.569Outbound Inter-processor Messages 25 ISH to SEC (ISH2SEC_MSG25)—Offset 1C0h ....

2312
20.570Outbound Inter-processor Messages 26 ISH to SEC (ISH2SEC_MSG26)—Offset 1C4h 2312
20.571Outbound Inter-processor Messages 27 ISH to SEC (ISH2SEC_MSG27)—Offset 1C8h 2313
20.572Outbound Inter-processor Messages 28 ISH to SEC (ISH2SEC_MSG28)—Offset 1CCh 2313
20.573Outbound Inter-processor Messages 29 ISH to SEC (ISH2SEC_MSG29)—Offset 1D0h 2314
20.574Outbound Inter-processor Messages 30 ISH to SEC (ISH2SEC_MSG30)—Offset 1D4h 2314
20.575Outbound Inter-processor Messages 31 ISH to SEC (ISH2SEC_MSG31)—Offset 1D8h 2314
20.576Outbound Inter-processor Messages 32 ISH to SEC (ISH2SEC_MSG32)—Offset 1DCh 2315
20.577Inbound Inter-processor Messages 1 SEC to ISH (SEC2ISH_MSG1)—Offset 1E0h ...2315
20.578Inbound Inter-processor Messages 2 SEC to ISH (SEC2ISH_MSG2)—Offset 1E4h ...2316
20.579Inbound Inter-processor Messages 3 SEC to ISH (SEC2ISH_MSG3)—Offset 1E8h ...2316
20.580Inbound Inter-processor Messages 4 SEC to ISH (SEC2ISH_MSG4)—Offset 1ECh ...2316
20.581Inbound Inter-processor Messages 5 SEC to ISH (SEC2ISH_MSG5)—Offset 1F0h ...2317
20.582Inbound Inter-processor Messages 6 SEC to ISH (SEC2ISH_MSG6)—Offset 1F4h ...2317
20.583Inbound Inter-processor Messages 7 SEC to ISH (SEC2ISH_MSG7)—Offset 1F8h ...2318
20.584Inbound Inter-processor Messages 8 SEC to ISH (SEC2ISH_MSG8)—Offset 1FCh ...2318
20.585Inbound Inter-processor Messages 9 SEC to ISH (SEC2ISH_MSG9)—Offset 200h ...2318
20.586Inbound Inter-processor Messages 10 SEC to ISH (SEC2ISH_MSG10)—Offset 204h2319
20.587Inbound Inter-processor Messages 11 SEC to ISH (SEC2ISH_MSG11)—Offset 208h2319
20.588Inbound Inter-processor Messages 12 SEC to ISH (SEC2ISH_MSG12)—Offset 20Ch2320
20.589Inbound Inter-processor Messages 13 SEC to ISH (SEC2ISH_MSG13)—Offset 210h2320
20.590Inbound Inter-processor Messages 14 SEC to ISH (SEC2ISH_MSG14)—Offset 214h2320
20.591Inbound Inter-processor Messages 15 SEC to ISH (SEC2ISH_MSG15)—Offset 218h2321
20.592Inbound Inter-processor Messages 16 SEC to ISH (SEC2ISH_MSG16)—Offset 21Ch2321
20.593Inbound Inter-processor Messages 17 SEC to ISH (SEC2ISH_MSG17)—Offset 220h2322
20.594Inbound Inter-processor Messages 18 SEC to ISH (SEC2ISH_MSG18)—Offset 224h2322
20.595Inbound Inter-processor Messages 19 SEC to ISH (SEC2ISH_MSG19)—Offset 228h2322
20.596Inbound Inter-processor Messages 20 SEC to ISH (SEC2ISH_MSG20)—Offset 22Ch2323
20.597Inbound Inter-processor Messages 21 SEC to ISH (SEC2ISH_MSG21)—Offset 230h2323
20.598Inbound Inter-processor Messages 22 SEC to ISH (SEC2ISH_MSG22)—Offset 234h2324
20.599Inbound Inter-processor Messages 23 SEC to ISH (SEC2ISH_MSG23)—Offset 238h2324
20.600Inbound Inter-processor Messages 24 SEC to ISH (SEC2ISH_MSG24)—Offset 23Ch2324
20.601Inbound Inter-processor Messages 25 SEC to ISH (SEC2ISH_MSG25)—Offset 240h2325
20.602Inbound Inter-processor Messages 26 SEC to ISH (SEC2ISH_MSG26)—Offset 244h2325
20.603Inbound Inter-processor Messages 27 SEC to ISH (SEC2ISH_MSG27)—Offset 248h2326
20.604Inbound Inter-processor Messages 28 SEC to ISH (SEC2ISH_MSG28)—Offset 24Ch2326
20.605Inbound Inter-processor Messages 29 SEC to ISH (SEC2ISH_MSG29)—Offset 250h2326
20.606Inbound Inter-processor Messages 30 SEC to ISH (SEC2ISH_MSG30)—Offset 254h2327
20.607Inbound Inter-processor Messages 31 SEC to ISH (SEC2ISH_MSG31)—Offset 258h2327
20.608Inbound Inter-processor Messages 32 SEC to ISH (SEC2ISH_MSG32)—Offset 25Ch2328
20.609Outbound Inter-processor Messages 1 ISH to PMC (ISH2PMC_MSG1)—Offset 260h 2328
20.610Outbound Inter-processor Messages 2 ISH to PMC (ISH2PMC_MSG2)—Offset 264h 2328
20.611Outbound Inter-processor Messages 3 ISH to PMC (ISH2PMC_MSG3)—Offset 268h 2329
20.612Outbound Inter-processor Messages 4 ISH to PMC (ISH2PMC_MSG4)—Offset 26Ch 2329
20.613Outbound Inter-processor Messages 5 ISH to PMC (ISH2PMC_MSG5)—Offset 270h 2330
20.614Outbound Inter-processor Messages 6 ISH to PMC (ISH2PMC_MSG6)—Offset 274h 2330
20.615Outbound Inter-processor Messages 7 ISH to PMC (ISH2PMC_MSG7)—Offset 278h 2330
20.616Outbound Inter-processor Messages 8 ISH to PMC (ISH2PMC_MSG8)—Offset 27Ch 2331

20.617Outbound Inter-processor Messages 9 ISH to PMC (ISH2PMC_MSG9)—Offset 280h 2331
20.618Outbound Inter-processor Messages 10 ISH to PMC (ISH2PMC_MSG10)—Offset 284h ... 2332
20.619Outbound Inter-processor Messages 11 ISH to PMC (ISH2PMC_MSG11)—Offset 288h ... 2332
20.620Outbound Inter-processor Messages 12 ISH to PMC (ISH2PMC_MSG12)—Offset 28Ch ... 2332
20.621Outbound Inter-processor Messages 13 ISH to PMC (ISH2PMC_MSG13)—Offset 290h ... 2333
20.622Outbound Inter-processor Messages 14 ISH to PMC (ISH2PMC_MSG14)—Offset 294h ... 2333
20.623Outbound Inter-processor Messages 15 ISH to PMC (ISH2PMC_MSG15)—Offset 298h ... 2334
20.624Outbound Inter-processor Messages 16 ISH to PMC (ISH2PMC_MSG16)—Offset 29Ch ... 2334
20.625Outbound Inter-processor Messages 17 ISH to PMC (ISH2PMC_MSG17)—Offset 2A0h ... 2334
20.626Outbound Inter-processor Messages 18 ISH to PMC (ISH2PMC_MSG18)—Offset 2A4h ... 2335
20.627Outbound Inter-processor Messages 19 ISH to PMC (ISH2PMC_MSG19)—Offset 2A8h ... 2335
20.628Outbound Inter-processor Messages 20 ISH to PMC (ISH2PMC_MSG20)—Offset 2ACh ... 2336
20.629Outbound Inter-processor Messages 21 ISH to PMC (ISH2PMC_MSG21)—Offset 2B0h ... 2336
20.630Outbound Inter-processor Messages 22 ISH to PMC (ISH2PMC_MSG22)—Offset 2B4h ... 2336
20.631Outbound Inter-processor Messages 23 ISH to PMC (ISH2PMC_MSG23)—Offset 2B8h ... 2337
20.632Outbound Inter-processor Messages 24 ISH to PMC (ISH2PMC_MSG24)—Offset 2BCh ... 2337
20.633Outbound Inter-processor Messages 25 ISH to PMC (ISH2PMC_MSG25)—Offset 2C0h ... 2338
20.634Outbound Inter-processor Messages 26 ISH to PMC (ISH2PMC_MSG26)—Offset 2C4h ... 2338
20.635Outbound Inter-processor Messages 27 ISH to PMC (ISH2PMC_MSG27)—Offset 2C8h ... 2338
20.636Outbound Inter-processor Messages 28 ISH to PMC (ISH2PMC_MSG28)—Offset 2CCh ... 2339
20.637Outbound Inter-processor Messages 29 ISH to PMC (ISH2PMC_MSG29)—Offset 2D0h ... 2339
20.638Outbound Inter-processor Messages 30 ISH to PMC (ISH2PMC_MSG30)—Offset 2D4h ... 2340
20.639Outbound Inter-processor Messages 31 ISH to PMC (ISH2PMC_MSG31)—Offset 2D8h ... 2340
20.640Outbound Inter-processor Messages 32 ISH to PMC (ISH2PMC_MSG32)—Offset 2DCh ... 2340
20.641Inbound Inter-processor Messages 1 PMC to ISH (PMC2ISH_MSG1)—Offset 2E0h .. 2341
20.642Inbound Inter-processor Messages 2 PMC to ISH (PMC2ISH_MSG2)—Offset 2E4h .. 2341
20.643Inbound Inter-processor Messages 3 PMC to ISH (PMC2ISH_MSG3)—Offset 2E8h .. 2342
20.644Inbound Inter-processor Messages 4 PMC to ISH (PMC2ISH_MSG4)—Offset 2ECh .. 2342
20.645Inbound Inter-processor Messages 5 PMC to ISH (PMC2ISH_MSG5)—Offset 2F0h .. 2342
20.646Inbound Inter-processor Messages 6 PMC to ISH (PMC2ISH_MSG6)—Offset 2F4h .. 2343
20.647Inbound Inter-processor Messages 7 PMC to ISH (PMC2ISH_MSG7)—Offset 2F8h .. 2343
20.648Inbound Inter-processor Messages 8 PMC to ISH (PMC2ISH_MSG8)—Offset 2FCh .. 2344
20.649Inbound Inter-processor Messages 9 PMC to ISH (PMC2ISH_MSG9)—Offset 300h .. 2344
20.650Inbound Inter-processor Messages 10 PMC to ISH (PMC2ISH_MSG10)—Offset 304h .....

2344
20.651Inbound Inter-processor Messages 11 PMC to ISH (PMC2ISH_MSG11)—Offset 308h 2345
20.652Inbound Inter-processor Messages 12 PMC to ISH (PMC2ISH_MSG12)—Offset 30Ch 2345
20.653Inbound Inter-processor Messages 13 PMC to ISH (PMC2ISH_MSG13)—Offset 310h 2346
20.654Inbound Inter-processor Messages 14 PMC to ISH (PMC2ISH_MSG14)—Offset 314h 2346
20.655Inbound Inter-processor Messages 15 PMC to ISH (PMC2ISH_MSG15)—Offset 318h 2346
20.656Inbound Inter-processor Messages 16 PMC to ISH (PMC2ISH_MSG16)—Offset 31Ch 2347
20.657Inbound Inter-processor Messages 17 PMC to ISH (PMC2ISH_MSG17)—Offset 320h 2347
20.658Inbound Inter-processor Messages 18 PMC to ISH (PMC2ISH_MSG18)—Offset 324h 2348
20.659Inbound Inter-processor Messages 19 PMC to ISH (PMC2ISH_MSG19)—Offset 328h 2348
20.660Inbound Inter-processor Messages 20 PMC to ISH (PMC2ISH_MSG20)—Offset 32Ch 2348
20.661Inbound Inter-processor Messages 21 PMC to ISH (PMC2ISH_MSG21)—Offset 330h 2349
20.662Inbound Inter-processor Messages 22 PMC to ISH (PMC2ISH_MSG22)—Offset 334h 2349
20.663Inbound Inter-processor Messages 23 PMC to ISH (PMC2ISH_MSG23)—Offset 338h 2350
20.664Inbound Inter-processor Messages 24 PMC to ISH (PMC2ISH_MSG24)—Offset 33Ch 2350
20.665Inbound Inter-processor Messages 25 PMC to ISH (PMC2ISH_MSG25)—Offset 340h 2350
20.666Inbound Inter-processor Messages 26 PMC to ISH (PMC2ISH_MSG26)—Offset 344h 2351
20.667Inbound Inter-processor Messages 27 PMC to ISH (PMC2ISH_MSG27)—Offset 348h 2351
20.668Inbound Inter-processor Messages 28 PMC to ISH (PMC2ISH_MSG28)—Offset 34Ch 2352
20.669Inbound Inter-processor Messages 29 PMC to ISH (PMC2ISH_MSG29)—Offset 350h 2352
20.670Inbound Inter-processor Messages 30 PMC to ISH (PMC2ISH_MSG30)—Offset 354h 2352
20.671Inbound Inter-processor Messages 31 PMC to ISH (PMC2ISH_MSG31)—Offset 358h 2353
20.672Inbound Inter-processor Messages 32 PMC to ISH (PMC2ISH_MSG32)—Offset 35Ch 2353
20.673Remap 0 (REMAP0)—Offset 360h 2353
20.674Remap 1 (REMAP1)—Offset 364h 2354
20.675Remap 2 (REMAP2)—Offset 368h 2354
20.676Remap 3 (REMAP3)—Offset 36Ch 2354
20.677Remap 4 (REMAP4)—Offset 370h 2355
20.678Remap 5 (REMAP5)—Offset 374h 2355
20.679ISH IPC Busy Clear (ISH_IPC_BUSY_CLEAR)—Offset 378h 2355
20.680ISH Fuse (ISH_FUSE)—Offset 37Ch 2356
20.681UMA Base Address LSB (UMA_RANGE_LOWER_0)—Offset 380h 2357
20.682UMA Base Address MSB (UMA_RANGE_LOWER_1)—Offset 384h 2357
20.683UMA Limit Address LSB (UMA_RANGE_UPPER_0)—Offset 388h 2357
20.684UMA Base Address MSB (UMA_RANGE_UPPER_1)—Offset 38Ch 2358

20.685Outbound Inter-processor Messages 1 ISH to ISP (ISH2ISP_MSG1)—Offset 390h .. 2358
20.686Outbound Inter-processor Messages 2 ISH to ISP (ISH2ISP_MSG2)—Offset 394h .. 2359
20.687Outbound Inter-processor Messages 3 ISH to ISP (ISH2ISP_MSG3)—Offset 398h .. 2359
20.688Outbound Inter-processor Messages 4 ISH to ISP (ISH2ISP_MSG4)—Offset 39Ch .. 2360
20.689Outbound Inter-processor Messages 5 ISH to ISP (ISH2ISP_MSG5)—Offset 3A0h .. 2360
20.690Outbound Inter-processor Messages 6 ISH to ISP (ISH2ISP_MSG6)—Offset 3A4h .. 2360
20.691Outbound Inter-processor Messages 7 ISH to ISP (ISH2ISP_MSG7)—Offset 3A8h .. 2361
20.692Outbound Inter-processor Messages 8 ISH to ISP (ISH2ISP_MSG8)—Offset 3ACh .. 2361
20.693Outbound Inter-processor Messages 9 ISH to ISP (ISH2ISP_MSG9)—Offset 3B0h .. 2362
20.694Outbound Inter-processor Messages 10 ISH to ISP (ISH2ISP_MSG10)—Offset 3B4h..... 2362
20.695Outbound Inter-processor Messages 11 ISH to ISP (ISH2ISP_MSG11)—Offset 3B8h..... 2363
20.696Outbound Inter-processor Messages 12 ISH to ISP (ISH2ISP_MSG12)—Offset 3BCh 2363
20.697Outbound Inter-processor Messages 13 ISH to ISP (ISH2ISP_MSG13)—Offset 3C0h..... 2363
20.698Outbound Inter-processor Messages 14 ISH to ISP (ISH2ISP_MSG14)—Offset 3C4h..... 2364
20.699Outbound Inter-processor Messages 15 ISH to ISP (ISH2ISP_MSG15)—Offset 3C8h..... 2364
20.700Outbound Inter-processor Messages 16 ISH to ISP (ISH2ISP_MSG16)—Offset 3CCh 2365
20.701Outbound Inter-processor Messages 17 ISH to ISP (ISH2ISP_MSG17)—Offset 3D0h 2365
20.702Outbound Inter-processor Messages 18 ISH to ISP (ISH2ISP_MSG18)—Offset 3D4h 2366
20.703Outbound Inter-processor Messages 19 ISH to ISP (ISH2ISP_MSG19)—Offset 3D8h 2366
20.704Outbound Inter-processor Messages 20 ISH to ISP (ISH2ISP_MSG20)—Offset 3DCh 2366
20.705Outbound Inter-processor Messages 21 ISH to ISP (ISH2ISP_MSG21)—Offset 3E0h..... 2367
20.706Outbound Inter-processor Messages 22 ISH to ISP (ISH2ISP_MSG22)—Offset 3E4h..... 2367
20.707Outbound Inter-processor Messages 23 ISH to ISP (ISH2ISP_MSG23)—Offset 3E8h..... 2368
20.708Outbound Inter-processor Messages 24 ISH to ISP (ISH2ISP_MSG24)—Offset 3ECh..... 2368
20.709Outbound Inter-processor Messages 25 ISH to ISP (ISH2ISP_MSG25)—Offset 3F0h..... 2369
20.710Outbound Inter-processor Messages 26 ISH to ISP (ISH2ISP_MSG26)—Offset 3F4h..... 2369
20.711Outbound Inter-processor Messages 27 ISH to ISP (ISH2ISP_MSG27)—Offset 3F8h..... 2369
20.712Outbound Inter-processor Messages 28 ISH to ISP (ISH2ISP_MSG28)—Offset 3FCh..... 2370
20.713Outbound Inter-processor Messages 29 ISH to ISP (ISH2ISP_MSG29)—Offset 400h..... 2370
20.714Outbound Inter-processor Messages 30 ISH to ISP (ISH2ISP_MSG30)—Offset 404h..... 2371
20.715Outbound Inter-processor Messages 31 ISH to ISP (ISH2ISP_MSG31)—Offset 408h..... 2371
20.716Outbound Inter-processor Messages 32 ISH to ISP (ISH2ISP_MSG32)—Offset 40Ch..... 2372
20.717Inbound Inter-processor Messages 1 ISP to ISH (ISP2ISH_MSG1)—Offset 410h 2372
20.718Inbound Inter-processor Messages2 ISP to ISH (ISP2ISH_MSG2)—Offset 414h 2372

20.719Inbound Inter-processor Messages 3 ISP to ISH (ISP2ISH_MSG3)—Offset 418h 2373
20.720Inbound Inter-processor Messages 4 ISP to ISH (ISP2ISH_MSG4)—Offset 41Ch.....2373
20.721Inbound Inter-processor Messages 5 ISP to ISH (ISP2ISH_MSG5)—Offset 420h.....2374
20.722Inbound Inter-processor Messages 6 ISP to ISH (ISP2ISH_MSG6)—Offset 424h.....2374
20.723Inbound Inter-processor Messages 7 ISP to ISH (ISP2ISH_MSG7)—Offset 428h.....2375
20.724Inbound Inter-processor Messages 8 ISP to ISH (ISP2ISH_MSG8)—Offset 42Ch.....2375
20.725Inbound Inter-processor Messages 9 ISP to ISH (ISP2ISH_MSG9)—Offset 430h.....2375
20.726Inbound Inter-processor Messages 10 ISP to ISH (ISP2ISH_MSG10)—Offset 434h .2376
20.727Inbound Inter-processor Messages 11 ISP to ISH (ISP2ISH_MSG11)—Offset 438h .2376
20.728Inbound Inter-processor Messages 12 ISP to ISH (ISP2ISH_MSG12)—Offset 43Ch .2377
20.729Inbound Inter-processor Messages 13 ISP to ISH (ISP2ISH_MSG13)—Offset 440h .2377
20.730Inbound Inter-processor Messages 14 ISP to ISH (ISP2ISH_MSG14)—Offset 444h .2378
20.731Inbound Inter-processor Messages 15 ISP to ISH (ISP2ISH_MSG15)—Offset 448h .2378
20.732Inbound Inter-processor Messages 16 ISP to ISH (ISP2ISH_MSG16)—Offset 44Ch .2378
20.733Inbound Inter-processor Messages 17 ISP to ISH (ISP2ISH_MSG17)—Offset 450h .2379
20.734Inbound Inter-processor Messages 18 ISP to ISH (ISP2ISH_MSG18)—Offset 454h .2379
20.735Inbound Inter-processor Messages 19 ISP to ISH (ISP2ISH_MSG19)—Offset 458h .2380
20.736Inbound Inter-processor Messages 20 ISP to ISH (ISP2ISH_MSG20)—Offset 45Ch .2380
20.737Inbound Inter-processor Messages 21 ISP to ISH (ISP2ISH_MSG21)—Offset 460h .2381
20.738Inbound Inter-processor Messages 22 ISP to ISH (ISP2ISH_MSG22)—Offset 464h .2381
20.739Inbound Inter-processor Messages 23 ISP to ISH (ISP2ISH_MSG23)—Offset 468h .2381
20.740Inbound Inter-processor Messages 24 ISP to ISH (ISP2ISH_MSG24)—Offset 46Ch .2382
20.741Inbound Inter-processor Messages 25 ISP to ISH (ISP2ISH_MSG25)—Offset 470h .2382
20.742Inbound Inter-processor Messages 26 ISP to ISH (ISP2ISH_MSG26)—Offset 474h .2383
20.743Inbound Inter-processor Messages 27 ISP to ISH (ISP2ISH_MSG27)—Offset 478h .2383
20.744Inbound Inter-processor Messages 28 ISP to ISH (ISP2ISH_MSG28)—Offset 47Ch .2384
20.745Inbound Inter-processor Messages 29 ISP to ISH (ISP2ISH_MSG29)—Offset 480h .2384
20.746Inbound Inter-processor Messages 30 ISP to ISH (ISP2ISH_MSG30)—Offset 484h .2384
20.747Inbound Inter-processor Messages 31 ISP to ISH (ISP2ISH_MSG31)—Offset 488h .2385
20.748Inbound Inter-processor Messages 32 ISP to ISH (ISP2ISH_MSG32)—Offset 48Ch .2385
20.749Outbound Inter-processor Messages 1 ISH to Audio (ISH2AUDIO_MSG1)—Offset 490h .. 2386
20.750Outbound Inter-processor Messages 2 ISH to Audio (ISH2AUDIO_MSG2)—Offset 494h .. 2386
20.751Outbound Inter-processor Messages 3 ISH to Audio (ISH2AUDIO_MSG3)—Offset 498h .. 2387
20.752Outbound Inter-processor Messages 4 ISH to Audio (ISH2AUDIO_MSG4)—Offset 49Ch .. 2387
20.753Outbound Inter-processor Messages 5 ISH to Audio (ISH2AUDIO_MSG5)—Offset 4A0h .. 2387
20.754Outbound Inter-processor Messages 6 ISH to Audio (ISH2AUDIO_MSG6)—Offset 4A4h .. 2388
20.755Outbound Inter-processor Messages 7 ISH to Audio (ISH2AUDIO_MSG7)—Offset 4A8h .. 2388
20.756Outbound Inter-processor Messages 8 ISH to Audio (ISH2AUDIO_MSG8)—Offset 4ACh .. 2389
20.757Outbound Inter-processor Messages 9 ISH to Audio (ISH2AUDIO_MSG9)—Offset 4B0h .. 2389
20.758Outbound Inter-processor Messages 10 ISH to Audio (ISH2AUDIO_MSG10)—Offset 4B4h 2390
20.759Outbound Inter-processor Messages 11 ISH to Audio (ISH2AUDIO_MSG11)—Offset 4B8h 2390
20.760Outbound Inter-processor Messages 12 ISH to Audio (ISH2AUDIO_MSG12)—Offset 4BCh 2390
20.761Outbound Inter-processor Messages 13 ISH to Audio (ISH2AUDIO_MSG13)—Offset 4C0h 2391

20.762Outbound Inter-processor Messages 14 ISH to Audio (ISH2AUDIO_MSG14)—Offset 4C4h 2391
20.763Outbound Inter-processor Messages 15 ISH to Audio (ISH2AUDIO_MSG15)—Offset 4C8h 2392
20.764Outbound Inter-processor Messages 16 ISH to Audio (ISH2AUDIO_MSG16)—Offset 4CCh 2392
20.765Outbound Inter-processor Messages 17 ISH to Audio (ISH2AUDIO_MSG17)—Offset 4D0h 2393
20.766Outbound Inter-processor Messages 18 ISH to Audio (ISH2AUDIO_MSG18)—Offset 4D4h 2393
20.767Outbound Inter-processor Messages 19 ISH to Audio (ISH2AUDIO_MSG19)—Offset 4D8h 2393
20.768Outbound Inter-processor Messages 20 ISH to Audio (ISH2AUDIO_MSG20)—Offset 4DCh 2394
20.769Outbound Inter-processor Messages 21 ISH to Audio (ISH2AUDIO_MSG21)—Offset 4E0h 2394
20.770Outbound Inter-processor Messages 22 ISH to Audio (ISH2AUDIO_MSG22)—Offset 4E4h 2395
20.771Outbound Inter-processor Messages 23 ISH to Audio (ISH2AUDIO_MSG23)—Offset 4E8h 2395
20.772Outbound Inter-processor Messages 24 ISH to Audio (ISH2AUDIO_MSG24)—Offset 4ECh 2396
20.773Outbound Inter-processor Messages 25 ISH to Audio (ISH2AUDIO_MSG25)—Offset 4F0h 2396
20.774Outbound Inter-processor Messages 26 ISH to Audio (ISH2AUDIO_MSG26)—Offset 4F4h 2396
20.775Outbound Inter-processor Messages 27 ISH to Audio (ISH2AUDIO_MSG27)—Offset 4F8h 2397
20.776Outbound Inter-processor Messages 28 ISH to Audio (ISH2AUDIO_MSG28)—Offset 4FCh 2397
20.777Outbound Inter-processor Messages 29 ISH to Audio (ISH2AUDIO_MSG29)—Offset 500h 2398
20.778Outbound Inter-processor Messages 30 ISH to Audio (ISH2AUDIO_MSG30)—Offset 504h 2398
20.779Outbound Inter-processor Messages 31 ISH to Audio (ISH2AUDIO_MSG31)—Offset 508h 2399
20.780Outbound Inter-processor Messages 32 ISH to Audio (ISH2AUDIO_MSG32)—Offset 50Ch 2399
20.781Inbound Inter-processor Messages 1 Audio to ISH (AUDIO2ISH_MSG1)—Offset 510h.... 2399
20.782Inbound Inter-processor Messages 2 Audio to ISH (AUDIO2ISH_MSG2)—Offset 514h.... 2400
20.783Inbound Inter-processor Messages 3 Audio to ISH (AUDIO2ISH_MSG3)—Offset 518h.... 2400
20.784Inbound Inter-processor Messages 4 Audio to ISH (AUDIO2ISH_MSG4)—Offset 51Ch ... 2401
20.785Inbound Inter-processor Messages 5 Audio to ISH (AUDIO2ISH_MSG5)—Offset 520h.... 2401
20.786Inbound Inter-processor Messages 6 Audio to ISH (AUDIO2ISH_MSG6)—Offset 524h.... 2402
20.787Inbound Inter-processor Messages 7 Audio to ISH (AUDIO2ISH_MSG7)—Offset 528h.... 2402
20.788Inbound Inter-processor Messages 8 Audio to ISH (AUDIO2ISH_MSG8)—Offset 52Ch ... 2402
20.789Inbound Inter-processor Messages 9 Audio to ISH (AUDIO2ISH_MSG9)—Offset 530h.... 2403
20.790Inbound Inter-processor Messages 10 Audio to ISH (AUDIO2ISH_MSG10)—Offset 534h

2403
20.791Inbound Inter-processor Messages 11 Audio to ISH (AUDIO2ISH_MSG11)—Offset 538h. 2404
20.792Inbound Inter-processor Messages 12 Audio to ISH (AUDIO2ISH_MSG12)—Offset 53Ch 2404
20.793Inbound Inter-processor Messages 13 Audio to ISH (AUDIO2ISH_MSG13)—Offset 540h. 2405
20.794Inbound Inter-processor Messages 14 Audio to ISH (AUDIO2ISH_MSG14)—Offset 544h. 2405
20.795Inbound Inter-processor Messages 15 Audio to ISH (AUDIO2ISH_MSG15)—Offset 548h. 2405
20.796Inbound Inter-processor Messages 16 Audio to ISH (AUDIO2ISH_MSG16)—Offset 54Ch 2406
20.797Inbound Inter-processor Messages 17 Audio to ISH (AUDIO2ISH_MSG17)—Offset 550h. 2406
20.798Inbound Inter-processor Messages 18 Audio to ISH (AUDIO2ISH_MSG18)—Offset 554h. 2407
20.799Inbound Inter-processor Messages 19 Audio to ISH (AUDIO2ISH_MSG19)—Offset 558h. 2407
20.800Inbound Inter-processor Messages 20 Audio to ISH (AUDIO2ISH_MSG20)—Offset 55Ch 2408
20.801Inbound Inter-processor Messages 21 Audio to ISH (AUDIO2ISH_MSG21)—Offset 560h. 2408
20.802Inbound Inter-processor Messages 22 Audio to ISH (AUDIO2ISH_MSG22)—Offset 564h. 2408
20.803Inbound Inter-processor Messages 23 Audio to ISH (AUDIO2ISH_MSG23)—Offset 568h. 2409
20.804Inbound Inter-processor Messages 24 Audio to ISH (AUDIO2ISH_MSG24)—Offset 56Ch 2409
20.805Inbound Inter-processor Messages 25 Audio to ISH (AUDIO2ISH_MSG25)—Offset 570h. 2410
20.806Inbound Inter-processor Messages 26 Audio to ISH (AUDIO2ISH_MSG26)—Offset 574h. 2410
20.807Inbound Inter-processor Messages 27 Audio to ISH (AUDIO2ISH_MSG27)—Offset 578h. 2411
20.808Inbound Inter-processor Messages 28 Audio to ISH (AUDIO2ISH_MSG28)—Offset 57Ch 2411
20.809Inbound Inter-processor Messages 29 Audio to ISH (AUDIO2ISH_MSG29)—Offset 580h. 2411
20.810Inbound Inter-processor Messages 30 Audio to ISH (AUDIO2ISH_MSG30)—Offset 584h. 2412
20.811Inbound Inter-processor Messages 31 Audio to ISH (AUDIO2ISH_MSG31)—Offset 588h. 2412
20.812Inbound Inter-processor Messages 32 Audio to ISH (AUDIO2ISH_MSG32)—Offset 58Ch 2413
20.813Outbound Inter-processor Messages 1 ISH to GFX (ISH2GFX_MSG1)—Offset 590h .2413
20.814Outbound Inter-processor Messages 2 ISH to GFX (ISH2GFX_MSG2)—Offset 594h .2414
20.815Outbound Inter-processor Messages 3 ISH to GFX (ISH2GFX_MSG3)—Offset 598h .2414
20.816Outbound Inter-processor Messages 4 ISH to GFX (ISH2GFX_MSG4)—Offset 59Ch .2414
20.817Outbound Inter-processor Messages 5 ISH to GFX (ISH2GFX_MSG5)—Offset 5A0h .2415
20.818Outbound Inter-processor Messages 6 ISH to GFX (ISH2GFX_MSG6)—Offset 5A4h .2415
20.819Outbound Inter-processor Messages 7 ISH to GFX (ISH2GFX_MSG7)—Offset 5A8h .2416
20.820Outbound Inter-processor Messages 8 ISH to GFX (ISH2GFX_MSG8)—Offset 5ACh .2416
20.821Outbound Inter-processor Messages 9 ISH to GFX (ISH2GFX_MSG9)—Offset 5B0h .2417
20.822Outbound Inter-processor Messages 10 ISH to GFX (ISH2GFX_MSG10)—Offset 5B4h 2417
20.823Outbound Inter-processor Messages 11 ISH to GFX (ISH2GFX_MSG11)—Offset 5B8h ....

2417
20.824Outbound Inter-processor Messages 12 ISH to GFX (ISH2GFX_MSG12)—Offset 5BCh....
2418
20.825Outbound Inter-processor Messages 13 ISH to GFX (ISH2GFX_MSG13)—Offset 5C0h....
2418
20.826Outbound Inter-processor Messages 14 ISH to GFX (ISH2GFX_MSG14)—Offset 5C4h....
2419
20.827Outbound Inter-processor Messages 15 ISH to GFX (ISH2GFX_MSG15)—Offset 5C8h....
2419
20.828Outbound Inter-processor Messages 16 ISH to GFX (ISH2GFX_MSG16)—Offset 5CCh....
2420
20.829Outbound Inter-processor Messages 17 ISH to GFX (ISH2GFX_MSG17)—Offset 5D0h....
2420
20.830Outbound Inter-processor Messages 18 ISH to GFX (ISH2GFX_MSG18)—Offset 5D4h....
2420
20.831Outbound Inter-processor Messages 19 ISH to GFX (ISH2GFX_MSG19)—Offset 5D8h....
2421
20.832Outbound Inter-processor Messages 20 ISH to GFX (ISH2GFX_MSG20)—Offset 5DCh ...
2421
20.833Outbound Inter-processor Messages 21 ISH to GFX (ISH2GFX_MSG21)—Offset 5E0h....
2422
20.834Outbound Inter-processor Messages 22 ISH to GFX (ISH2GFX_MSG22)—Offset 5E4h....
2422
20.835Outbound Inter-processor Messages 23 ISH to GFX (ISH2GFX_MSG23)—Offset 5E8h....
2423
20.836Outbound Inter-processor Messages 24 ISH to GFX (ISH2GFX_MSG24)—Offset 5ECh....
2423
20.837Outbound Inter-processor Messages 25 ISH to GFX (ISH2GFX_MSG25)—Offset 5F0h ....
2423
20.838Outbound Inter-processor Messages 26 ISH to GFX (ISH2GFX_MSG26)—Offset 5F4h ....
2424
20.839Outbound Inter-processor Messages 27 ISH to GFX (ISH2GFX_MSG27)—Offset 5F8h ....
2424
20.840Outbound Inter-processor Messages 28 ISH to GFX (ISH2GFX_MSG28)—Offset 5FCh....
2425
20.841Outbound Inter-processor Messages 29 ISH to GFX (ISH2GFX_MSG29)—Offset 600h....
2425
20.842Outbound Inter-processor Messages 30 ISH to GFX (ISH2GFX_MSG30)—Offset 604h....
2426
20.843Outbound Inter-processor Messages 31 ISH to GFX (ISH2GFX_MSG31)—Offset 608h....
2426
20.844Outbound Inter-processor Messages 32 ISH to GFX (ISH2GFX_MSG32)—Offset 60Ch....
2426
20.845Inbound Inter-processor Messages 1 GFX to ISH (GFX2ISH_MSG1)—Offset 610h .. 2427
20.846Inbound Inter-processor Messages 2 GFX to ISH (GFX2ISH_MSG2)—Offset 614h .. 2427
20.847Inbound Inter-processor Messages 3 GFX to ISH (GFX2ISH_MSG3)—Offset 618h .. 2428
20.848Inbound Inter-processor Messages 4 GFX to ISH (GFX2ISH_MSG4)—Offset 61Ch .. 2428
20.849Inbound Inter-processor Messages 5 GFX to ISH (GFX2ISH_MSG5)—Offset 620h .. 2429
20.850Inbound Inter-processor Messages 6 GFX to ISH (GFX2ISH_MSG6)—Offset 624h .. 2429
20.851Inbound Inter-processor Messages 7 GFX to ISH (GFX2ISH_MSG7)—Offset 628h .. 2429
20.852Inbound Inter-processor Messages 8 GFX to ISH (GFX2ISH_MSG8)—Offset 62Ch .. 2430
20.853Inbound Inter-processor Messages 9 GFX to ISH (GFX2ISH_MSG9)—Offset 630h .. 2430
20.854Inbound Inter-processor Messages 10 GFX to ISH (GFX2ISH_MSG10)—Offset 634h2431
20.855Inbound Inter-processor Messages 11 GFX to ISH (GFX2ISH_MSG11)—Offset 638h2431
20.856Inbound Inter-processor Messages 12 GFX to ISH (GFX2ISH_MSG12)—Offset 63Ch.....
2432
20.857Inbound Inter-processor Messages 13 GFX to ISH (GFX2ISH_MSG13)—Offset 640h2432

20.858Inbound Inter-processor Messages 14 GFX to ISH (GFX2ISH_MSG14)—Offset 644h2432
20.859Inbound Inter-processor Messages 15 GFX to ISH (GFX2ISH_MSG15)—Offset 648h2433
20.860Inbound Inter-processor Messages 16 GFX to ISH (GFX2ISH_MSG16)—Offset 64Ch 2433
20.861Inbound Inter-processor Messages 17 GFX to ISH (GFX2ISH_MSG17)—Offset 650h2434
20.862Inbound Inter-processor Messages 18 GFX to ISH (GFX2ISH_MSG18)—Offset 654h2434
20.863Inbound Inter-processor Messages 19 GFX to ISH (GFX2ISH_MSG19)—Offset 658h2435
20.864Inbound Inter-processor Messages 20 GFX to ISH (GFX2ISH_MSG20)—Offset 65Ch 2435
20.865Inbound Inter-processor Messages 21 GFX to ISH (GFX2ISH_MSG21)—Offset 660h2435
20.866Inbound Inter-processor Messages 22 GFX to ISH (GFX2ISH_MSG22)—Offset 664h2436
20.867Inbound Inter-processor Messages 23 GFX to ISH (GFX2ISH_MSG23)—Offset 668h2436
20.868Inbound Inter-processor Messages 24 GFX to ISH (GFX2ISH_MSG24)—Offset 66Ch 2437
20.869Inbound Inter-processor Messages 25 GFX to ISH (GFX2ISH_MSG25)—Offset 670h2437
20.870Inbound Inter-processor Messages 26 GFX to ISH (GFX2ISH_MSG26)—Offset 674h2438
20.871Inbound Inter-processor Messages 27 Gfx to ISH (GFX2ISH_MSG27)—Offset 678h 2438
20.872Inbound Inter-processor Messages 28 GFX to ISH (GFX2ISH_MSG28)—Offset 67Ch 2438
20.873Inbound Inter-processor Messages 29 GFX to ISH (GFX2ISH_MSG29)—Offset 680h2439
20.874Inbound Inter-processor Messages 30 GFX to ISH (GFX2ISH_MSG30)—Offset 684h2439
20.875Inbound Inter-processor Messages 31 GFX to ISH (GFX2ISH_MSG31)—Offset 688h2440
20.876Inbound Inter-processor Messages 32 GFX to ISH (GFX2ISH_MSG32)—Offset 68Ch 2440
20.877Inbound Doorbell ISP to ISH (ISP2ISH_DOORBELL)—Offset 690h.....2441
20.878Outbound Doorbell ISH to ISP (ISH2ISP_DOORBELL)—Offset 694h.....2441
20.879Inbound Doorbell Audio to ISH (AUDIO2ISH_DOORBELL)—Offset 698h 2441
20.880Outbound Doorbell ISH to Audio (ISH2AUDIO_DOORBELL)—Offset 69Ch.....2442
20.881Inbound Doorbell GFX to ISH (GFX2ISH_DOORBELL)—Offset 6A0h.....2442
20.882Outbound Doorbell ISH to GFX (ISH2GFX_DOORBELL)—Offset 6A4h.....2443
20.883CSME to ISH Control and Status (CSME2ISH_CTRL_STATUS)—Offset 6A8h 2443
20.884ISH IPC D0I3 Control (IPC_d0i3C_reg)—Offset 6D0h.....2444
20.885PMC to ISH Control and Status (ipc_pmc2ish_csr_reg)—Offset 6D4h 2444
20.886ISP to ISH Control and Status (ipc_isp2ish_csr_reg)—Offset 6D8h 2445
20.887Audio to ISH Control and Status (ipc_audio2ish_csr_reg)—Offset 6DCh.....2445
20.888RID - Bridge revision ID Register (RID)—Offset 30h 2445
20.889GEN_REGRW1 - General Purpose register (GEN_REGRW1)—Offset 600h 2446
20.890GEN_REGRW2 - General Purpose register (GEN_REGRW2)—Offset 604h 2446
20.891GEN_REGRW3 - General Purpose register (GEN_REGRW3)—Offset 608h 2446
20.892GEN_REGRW4 - General Purpose register (GEN_REGRW4)—Offset 60Ch 2447

21 HECI Registers 2448

21.1 Identifiers (HECI1_ID) — Offset 0h.....2448
21.2 Command (HECI1_CMD) — Offset 4h.....2448
21.3 Status (HECI1_STS) — Offset 6h.....2449
21.4 Revision ID And Class Code (HECI1_RID_CC) — Offset 8h 2450
21.5 Cache Line Size (HECI1_CLS) — Offset Ch.....2450
21.6 Master Latency Timer (HECI1_MLT) — Offset Dh 2450
21.7 Header Type (HECI1_HTYPE) — Offset Eh.....2451
21.8 Built In Self-Test (HECI1_BIST) — Offset Fh 2451
21.9 HECI MMIO Base Address Low (HECI1_MMIO_MBAR_LO) — Offset 10h.....2451
21.10 HECI MMIO Base Address High (HECI1_MMIO_MBAR_HI) — Offset 14h 2452
21.11 Sub System Identifiers (HECI1_SS) — Offset 2Ch.....2452
21.12 Capabilities Pointer (HECI1_CAP) — Offset 34h 2452
21.13 Interrupt Information (HECI1_INTR) — Offset 3Ch.....2453
21.14 Minimum Grant (HECI1_MGNT) — Offset 3Eh.....2453

21.15 Maximum Latency (HECI1_MLAT) — Offset 3Fh 2453
21.16 Host Firmware Status (HECI1_HFS) — Offset 40h 2454
21.17 Miscellaneous Shadow (HECI1_MISC_SHDW) — Offset 44h 2454
21.18 General Status Shadow 1 (HECI1_GS_SHDW1) — Offset 48h 2454
21.19 Host General Status (HECI1_H_GS1) — Offset 4Ch 2455
21.20 PCI Power Management Capability ID (HECI1_PID) — Offset 50h 2455
21.21 PCI Power Management Capabilities (HECI1_PC) — Offset 52h 2455
21.22 PCI Power Management Control And Status (HECI1_PMCS) — Offset 54h 2456
21.23 General Status Shadow 2 (HECI1_GS_SHDW2) — Offset 60h 2457
21.24 General Status Shadow 3 (HECI1_GS_SHDW3) — Offset 64h 2457
21.25 General Status Shadow 4 (HECI1_GS_SHDW4) — Offset 68h 2457
21.26 General Status Shadow 5 (HECI1_GS_SHDW5) — Offset 6Ch 2457
21.27 Host General Status 2 (HECI1_H_GS2) — Offset 70h 2458
21.28 Host General Status 3 (HECI1_H_GS3) — Offset 74h 2458
21.29 Message Signaled Interrupt Identifiers (HECI1_MID) — Offset 8Ch 2458
21.30 Message Signaled Interrupt Message Control (HECI1_MC) — Offset 8Eh 2459
21.31 Message Signaled Interrupt Message Address (HECI1_MA) — Offset 90h 2459
21.32 Message Signaled Interrupt Upper Address (HECI1_MUA) — Offset 94h 2459
21.33 Message Signaled Interrupt Message Data (HECI1_MD) — Offset 98h 2460
21.34 HECI Interrupt Delivery Mode (HECI1_HIDM) — Offset A0h 2460
21.35 Vendor Specific Capability Register (HECI1_VSCR) — Offset A4h 2460
21.36 Vendor Specific Extended Capability Register (HECI1_VSEC) — Offset A8h 2461
21.37 SW LTR Pointer Register (HECI1_SWLTRPTR) — Offset ACh 2461
21.38 Device Idle Pointer Register (HECI1_DEVIDLEPTR) — Offset B0h 2462
21.39 Device Idle Power On Latency (HECI1_DEVIDLEPOL) — Offset B4h 2462
21.40 DevIdle Power Control Enabled Register (HECI1_PWRCTRLEN) — Offset B6h 2462
21.41 Host Extend Register Status (HECI1_HERS) — Offset BCh 2463
21.42 Host Extend Register DW1 (HECI1_HER1) — Offset C0h 2463
21.43 Host Extend Register DW2 (HECI1_HER2) — Offset C4h 2464
21.44 Host Extend Register DW3 (HECI1_HER3) — Offset C8h 2464
21.45 Host Extend Register DW4 (HECI1_HER4) — Offset CCh 2464
21.46 Host Extend Register DW5 (HECI1_HER5) — Offset D0h 2464
21.47 Host Extend Register DW6 (HECI1_HER6) — Offset D4h 2465
21.48 Host Extend Register DW7 (HECI1_HER7) — Offset D8h 2465
21.49 Host Extend Register DW8 (HECI1_HER8) — Offset DCh 2465
21.50 Manufacturer's ID (HECI1_MANID) — Offset F8h 2465

22 HECI1 Registers 2467

22.1 Identifiers (HECI1_ID) — Offset 0h 2467
22.2 Command (HECI1_CMD) — Offset 4h 2467
22.3 Status (HECI1_STS) — Offset 6h 2468
22.4 Revision ID And Class Code (HECI1_RID_CC) — Offset 8h 2469
22.5 Cache Line Size (HECI1_CLS) — Offset Ch 2469
22.6 Master Latency Timer (HECI1_MLT) — Offset Dh 2469
22.7 Header Type (HECI1_HTYPE) — Offset Eh 2469
22.8 Built In Self-Test (HECI1_BIST) — Offset Fh 2470
22.9 HECI MMIO Base Address Low (HECI1_MMIO_MBAR_LO) — Offset 10h 2470
22.10 HECI MMIO Base Address High (HECI1_MMIO_MBAR_HI) — Offset 14h 2471
22.11 BAR1 — Offset 18h 2471
22.12 BAR1_HIGH — Offset 1Ch 2471
22.13 Sub System Identifiers (HECI1_SS) — Offset 2Ch 2471
22.14 EXPANSION_ROM_BASEADDR — Offset 30h 2472
22.15 Capabilities Pointer (HECI1_CAP) — Offset 34h 2472
22.16 Interrupt Information (HECI1_INTR) — Offset 3Ch 2472
22.17 Minimum Grant (HECI1_MGNT) — Offset 3Eh 2473

22.18 Maximum Latency (HECI1_MLAT) — Offset 3Fh 2473
22.19 Host Firmware Status (HECI1_HFS) — Offset 40h 2473
22.20 Miscellaneous Shadow (HECI1_MISC_SHDW) — Offset 44h 2474
22.21 General Status Shadow 1 (HECI1_GS_SHDW1) — Offset 48h 2474
22.22 Host General Status (HECI1_H_GS1) — Offset 4Ch 2474
22.23 PCI Power Management Capability ID (HECI1_PID) — Offset 50h 2475
22.24 PCI Power Management Capabilities (HECI1_PC) — Offset 52h 2475
22.25 PCI Power Management Control And Status (HECI1_PMCS) — Offset 54h 2476
22.26 General Status Shadow 2 (HECI1_GS_SHDW2) — Offset 60h 2476
22.27 General Status Shadow 3 (HECI1_GS_SHDW3) — Offset 64h 2477
22.28 General Status Shadow 4 (HECI1_GS_SHDW4) — Offset 68h 2477
22.29 General Status Shadow 5 (HECI1_GS_SHDW5) — Offset 6Ch 2477
22.30 Host General Status 2 (HECI1_H_GS2) — Offset 70h 2477
22.31 Host General Status 3 (HECI1_H_GS3) — Offset 74h 2478
22.32 POWERCAPID — Offset 80h 2478
22.33 PMECTRLSTATUS — Offset 84h 2478
22.34 Message Signaled Interrupt Identifiers (HECI1_MID) — Offset 8Ch 2479
22.35 Message Signaled Interrupt Message Control (HECI1_MC) — Offset 8Eh 2479
22.36 Message Signaled Interrupt Message Address (HECI1_MA) — Offset 90h 2479
22.37 Message Signaled Interrupt Upper Address (HECI1_MUA) — Offset 94h 2480
22.38 Message Signaled Interrupt Message Data (HECI1_MD) — Offset 98h 2480
22.39 DEVICE_IDLE_POINTER_REG — Offset 9Ch 2480
22.40 HECI Interrupt Delivery Mode (HECI1_HIDM) — Offset A0h 2481
22.41 Vendor Specific Capability Register (HECI1_VSCR) — Offset A4h 2481
22.42 Vendor Specific Extended Capability Register (HECI1_VSEC) — Offset A8h 2481
22.43 SW LTR Pointer Register (HECI1_SWLTRPTR) — Offset ACh 2482
22.44 Device Idle Pointer Register (HECI1_DEVIDLEPTR) — Offset B0h 2482
22.45 Device Idle Power On Latency (HECI1_DEVIDLEPOL) — Offset B4h 2483
22.46 DevIdle Power Control Enabled Register (HECI1_PWRCTRLEN) — Offset B6h 2483
22.47 GEN_REGRW3 — Offset B8h 2484
22.48 Host Extend Register Status (HECI1_HERS) — Offset BCh 2484
22.49 Host Extend Register DW1 (HECI1_HER1) — Offset C0h 2485
22.50 Host Extend Register DW2 (HECI1_HER2) — Offset C4h 2485
22.51 Host Extend Register DW3 (HECI1_HER3) — Offset C8h 2485
22.52 Host Extend Register DW4 (HECI1_HER4) — Offset CCh 2485
22.53 Host Extend Register DW5 (HECI1_HER5) — Offset D0h 2486
22.54 Host Extend Register DW6 (HECI1_HER6) — Offset D4h 2486
22.55 Host Extend Register DW7 (HECI1_HER7) — Offset D8h 2486
22.56 Host Extend Register DW8 (HECI1_HER8) — Offset DCh 2486
22.57 Manufacturer's ID (HECI1_MANID) — Offset F8h 2487

23 HECI2 Registers 2488

23.1 Identifiers (HECI2_ID) — Offset 0h 2488
23.2 Command (HECI2_CMD) — Offset 4h 2488
23.3 Status (HECI2_STS) — Offset 6h 2489
23.4 Revision ID And Class Code (HECI2_RID_CC) — Offset 8h 2490
23.5 Cache Line Size (HECI2_CLS) — Offset Ch 2490
23.6 Master Latency Timer (HECI2_MLT) — Offset Dh 2490
23.7 Header Type (HECI2_HTYPE) — Offset Eh 2490
23.8 Built In Self-Test (HECI2_BIST) — Offset Fh 2491
23.9 HECI MMIO Base Address Low (HECI2_MMIO_MBAR_LO) — Offset 10h 2491
23.10 HECI MMIO Base Address High (HECI2_MMIO_MBAR_HI) — Offset 14h 2492
23.11 Sub System Identifiers (HECI2_SS) — Offset 2Ch 2492
23.12 Capabilities Pointer (HECI2_CAP) — Offset 34h 2492
23.13 Interrupt Information (HECI2_INTR) — Offset 3Ch 2492

23.14 Minimum Grant (HECI2_MGNT) — Offset 3Eh 2493
23.15 Maximum Latency (HECI2_MLAT) — Offset 3Fh 2493
23.16 Host Firmware Status (HECI2_HFS) — Offset 40h 2493
23.17 General Status Shadow 1 (HECI2_GS_SHDW1) — Offset 48h 2494
23.18 Host General Status (HECI2_H_GS1) — Offset 4Ch 2494
23.19 PCI Power Management Capability ID (HECI2_PID) — Offset 50h 2494
23.20 PCI Power Management Capabilities (HECI2_PC) — Offset 52h 2494
23.21 PCI Power Management Control And Status (HECI2_PMCS) — Offset 54h 2495
23.22 General Status Shadow 2 (HECI2_GS_SHDW2) — Offset 60h 2496
23.23 General Status Shadow 3 (HECI2_GS_SHDW3) — Offset 64h 2496
23.24 General Status Shadow 4 (HECI2_GS_SHDW4) — Offset 68h 2496
23.25 General Status Shadow 5 (HECI2_GS_SHDW5) — Offset 6Ch 2497
23.26 Host General Status 2 (HECI2_H_GS2) — Offset 70h 2497
23.27 Host General Status 3 (HECI2_H_GS3) — Offset 74h 2497
23.28 Message Signaled Interrupt Identifiers (HECI2_MID) — Offset 8Ch 2497
23.29 Message Signaled Interrupt Message Control (HECI2_MC) — Offset 8Eh 2498
23.30 Message Signaled Interrupt Message Address (HECI2_MA) — Offset 90h 2498
23.31 Message Signaled Interrupt Upper Address (HECI2_MUA) — Offset 94h 2499
23.32 Message Signaled Interrupt Message Data (HECI2_MD) — Offset 98h 2499
23.33 HECI Interrupt Delivery Mode (HECI2_HIDM) — Offset A0h 2499
23.34 Vendor Specific Capability Register (HECI2_VSCR) — Offset A4h 2500
23.35 Vendor Specific Extended Capability Register (HECI2_VSEC) — Offset A8h 2500
23.36 SW LTR Pointer Register (HECI2_SWLTRPTR) — Offset ACh 2500
23.37 Device Idle Pointer Register (HECI2_DEVIDLEPTR) — Offset B0h 2501
23.38 Device Idle Power On Latency (HECI2_DEVIDLEPOL) — Offset B4h 2501
23.39 DevIdle Power Control Enabled Register (HECI2_PWRCTRLEN) — Offset B6h 2502
23.40 Manufacturer's ID (HECI2_MANID) — Offset F8h 2502

24 HECI3 Registers 2504

24.1 Identifiers (HECI3_ID) — Offset 0h 2504
24.2 Command (HECI3_CMD) — Offset 4h 2504
24.3 Status (HECI3_STS) — Offset 6h 2505
24.4 Revision ID And Class Code (HECI3_RID_CC) — Offset 8h 2506
24.5 Cache Line Size (HECI3_CLS) — Offset Ch 2506
24.6 Master Latency Timer (HECI3_MLT) — Offset Dh 2506
24.7 Header Type (HECI3_HTYPE) — Offset Eh 2506
24.8 Built In Self-Test (HECI3_BIST) — Offset Fh 2507
24.9 HECI MMIO Base Address Low (HECI3_MMIO_MBAR_LO) — Offset 10h 2507
24.10 HECI MMIO Base Address High (HECI3_MMIO_MBAR_HI) — Offset 14h 2508
24.11 Sub System Identifiers (HECI3_SS) — Offset 2Ch 2508
24.12 Capabilities Pointer (HECI3_CAP) — Offset 34h 2508
24.13 Interrupt Information (HECI3_INTR) — Offset 3Ch 2508
24.14 Minimum Grant (HECI3_MGNT) — Offset 3Eh 2509
24.15 Maximum Latency (HECI3_MLAT) — Offset 3Fh 2509
24.16 Host Firmware Status (HECI3_HFS) — Offset 40h 2509
24.17 General Status Shadow 1 (HECI3_GS_SHDW1) — Offset 48h 2510
24.18 Host General Status (HECI3_H_GS1) — Offset 4Ch 2510
24.19 PCI Power Management Capability ID (HECI3_PID) — Offset 50h 2510
24.20 PCI Power Management Capabilities (HECI3_PC) — Offset 52h 2510
24.21 PCI Power Management Control And Status (HECI3_PMCS) — Offset 54h 2511
24.22 General Status Shadow 2 (HECI3_GS_SHDW2) — Offset 60h 2512
24.23 General Status Shadow 3 (HECI3_GS_SHDW3) — Offset 64h 2512
24.24 General Status Shadow 4 (HECI3_GS_SHDW4) — Offset 68h 2512
24.25 General Status Shadow 5 (HECI3_GS_SHDW5) — Offset 6Ch 2513
24.26 Host General Status 2 (HECI3_H_GS2) — Offset 70h 2513

24.27 Host General Status 3 (HECI3_H_GS3) — Offset 74h 2513
24.28 Message Signaled Interrupt Identifiers (HECI3_MID) — Offset 8Ch....2513
24.29 Message Signaled Interrupt Message Control (HECI3_MC) — Offset 8Eh 2514
24.30 Message Signaled Interrupt Message Address (HECI3_MA) — Offset 90h 2514
24.31 Message Signaled Interrupt Upper Address (HECI3_MUA) — Offset 94h 2515
24.32 Message Signaled Interrupt Message Data (HECI3_MD) — Offset 98h....2515
24.33 HECI Interrupt Delivery Mode (HECI3_HIDM) — Offset A0h....2515
24.34 Vendor Specific Capability Register (HECI3_VSCR) — Offset A4h 2516
24.35 Vendor Specific Extended Capability Register (HECI3_VSEC) — Offset A8h....2516
24.36 SW LTR Pointer Register (HECI3_SWLTRPTR) — Offset ACh 2516
24.37 Device Idle Pointer Register (HECI3_DEVIDLEPTR) — Offset B0h 2517
24.38 Device Idle Power On Latency (HECI3_DEVIDLEPOL) — Offset B4h 2517
24.39 DevIdle Power Control Enabled Register (HECI3_PWRCTRLEN) — Offset B6h....2518
24.40 Manufacturer's ID (HECI3_MANID) — Offset F8h 2518

25 HD Audio Registers 2520

25.1 VID_DID — Offset 0h....2520
25.2 CMD_STS — Offset 4h 2520
25.3 RID_PI_SCC_BCC — Offset 8h 2521
25.4 CLS_LT_HTYPE_BIST — Offset Ch....2522
25.5 Intel HD Audio Base Lower Address (HDALBA) — Offset 10h 2522
25.6 Intel HD Audio Base Upper Address (HDAUBA) — Offset 14h....2523
25.7 Shadowed PCI Configuration Lower Base Address (SPCLBA) — Offset 18h....2523
25.8 Shadowed PCI Configuration Upper Base Address (SPCUBA) — Offset 1Ch 2524
25.9 Audio DSP Lower Base Address (ADSPLBA) — Offset 20h....2524
25.10 Audio DSP Upper Base Address (ADSPUBA) — Offset 24h 2525
25.11 SVID_SID — Offset 2Ch 2525
25.12 INTLN_INTPN — Offset 3Ch 2525
25.13 PID_PC — Offset 50h 2526
25.14 Power Management Control And Status (PCS) — Offset 54h 2526
25.15 MID_MMC — Offset 60h 2528
25.16 MSI Message Lower Address (MMLA) — Offset 64h 2528
25.17 MSI Message Upper Address (MMUA) — Offset 68h....2528
25.18 MSI Message Data (MMD) — Offset 6Ch 2529
25.19 (GCAP_VMIN_VMAJ)—Offset 0h....2529
25.20 (OUTPAY_INPAY)—Offset 4h 2530
25.21 Global Control (GCTL)—Offset 8h....2531
25.22 (WAKEEN_WAKESTS)—Offset Ch....2532
25.23 (GSTS_GCAP2)—Offset 10h 2533
25.24 Linked List Capabilities Header (LLCH)—Offset 14h 2533
25.25 (OUTSTRMPAY_INSTRMPAY)—Offset 18h....2534
25.26 Interrupt Control (INTCTL)—Offset 20h 2535
25.27 Interrupt Status (INTSTS)—Offset 24h 2536
25.28 Wall Clock Counter (WALCLK)—Offset 30h 2537
25.29 Stream Synchronization (SSYNC)—Offset 38h 2537
25.30 CORB Lower Base Address (CORBLBASE)—Offset 40h 2538
25.31 CORB Upper Base Address (CORBUBASE)—Offset 44h....2538
25.32 (CORBWP_CORBRP)—Offset 48h 2539
25.33 (CORBCTL_CORBSTS_CORBSIZE)—Offset 4Ch 2539
25.34 RIRB Lower Base Address (RIRBLBASE)—Offset 50h 2540
25.35 RIRB Upper Base Address (RIRBUBASE)—Offset 54h....2541
25.36 (RIRBWP_RINTCNT)—Offset 58h 2541
25.37 (RIRBCTL_RIRBSTS_RIRBSIZE)—Offset 5Ch 2542
25.38 Immediate Command (IC)—Offset 60h....2543
25.39 Immediate Response (IR)—Offset 64h 2543

25.40 Immediate Command Status (ICS)—Offset 68h 2544
25.41 DMA Position Lower Base Address (DPLBASE)—Offset 70h 2545
25.42 DMA Position Upper Base Address (DPUBASE)—Offset 74h 2545
25.43 (ISD0CTL_ISD0STS)—Offset 80h 2545
25.44 Input/Output Stream Descriptor x Link Position in Buffer (ISD0LPIB)—Offset 84h ... 2548
25.45 Input/Output Stream Descriptor x Cyclic Buffer Length (ISD0CBL)—Offset 88h 2549
25.46 (ISD0LVI_ISD0FIFOW)—Offset 8Ch 2549

26 SRAM Registers 2550

26.1 DEVICEVENDORID - Device ID and Vendor ID Register (DEVVENDID) — Offset 0h . 2550
26.2 STATUSCOMMAND — Offset 4h 2550
26.3 REVCLASSCODE — Offset 8h 2551
26.4 CLLATHEADERBIST — Offset Ch 2551
26.5 BAR — Offset 10h 2552
26.6 BAR -Base Address Register High (BAR_HIGH) — Offset 14h 2552
26.7 BAR1 — Offset 18h 2553
26.8 BAR1 -Base Address Register1 High (BAR1_HIGH) — Offset 1Ch 2553
26.9 BAR2 — Offset 20h 2554
26.10 SUBSYSTEMID — Offset 2Ch 2554
26.11 EXPANSION ROM base address (EXPANSION_ROM_BASEADDR) — Offset 30h 2554
26.12 CAPABILITYPTR — Offset 34h 2555
26.13 INTERRUPTREG — Offset 3Ch 2555
26.14 POWERCAPID — Offset 80h 2555
26.15 PMECTRLSTATUS — Offset 84h 2556
26.16 PCI DEVICE IDLE CAPABILITY RECORD (PCIDEVIDLE_CAP_RECORD) — Offset 90h. 2556
26.17 DEVID VENDOR SPECIFIC REG (DEVID_VEND_SPECIFIC_REG) — Offset 94h 2557
26.18 D0I3_CONTROL_SW_LTR_MMIO_REG — Offset 98h 2557
26.19 DEVICE_IDLE_POINTER_REG — Offset 9Ch 2558
26.20 D0I3_MAX_POW_LAT_PG_CONFIG — Offset A0h 2558
26.21 GEN_REGRW1 - General Purpose Read Write Register1 (GEN_REGRW1) — Offset B0h ... 2559
26.22 GEN_REGRW2 — Offset B4h 2559
26.23 GEN_REGRW3 — Offset B8h 2559
26.24 GEN_REGRW4 — Offset BCh 2560
26.25 GEN_INPUT_REG — Offset C0h 2560
26.26 Manufacturers ID (MANID) — Offset F8h 2560

27 SPI Controller Registers 2562

27.1 Device ID and Vendor ID (BIOS_SPI_DID_VID) — Offset 0h 2562
27.2 Status and Command (BIOS_SPI_STS_CMD) — Offset 4h 2562
27.3 Revision ID and Class Code (BIOS_SPI_CC_RID) — Offset 8h 2563
27.4 BIST, Header Type, Latency Timer, Cache Line Size (BIOS_SPI_BIST_HTYPE_LT_CLS) — Offset Ch 2564
27.5 SPI BAR0 MMIO (BIOS_SPI_BAR0) — Offset 10h 2564
27.6 CSXE BAR1 Direct Reads (CSXE_SPI_BAR1) — Offset 14h 2565
27.7 CSXE BAR2 Huffman Decompression (CSXE_SPI_BAR2) — Offset 18h 2565
27.8 CSXE BAR3 Touch BAR (CSXE_SPI_BAR3) — Offset 1Ch 2566
27.9 CSXE BAR4 BIOS Direct Reads (CSXE_SPI_BAR4) — Offset 20h 2566
27.10 Cardbus CIS Pointer (CSXE_SPI_CCP) — Offset 28h 2567
27.11 Subsystem and Vendor ID (BIOS_SPI_SID_SVID) — Offset 2Ch 2567
27.12 Expansion ROM Base Address (CSXE_SPI_XRBAR) — Offset 30h 2567
27.13 Capabilities List Pointer (CSXE_SPI_CAPP) — Offset 34h 2568
27.14 Maximum Latency, Minimum Grant, Interrupt Pin and Interrupt Line (CSXE_SPI_MAX-L_MING_INTP_INTL) — Offset 3Ch 2568
27.15 CSXE Capabilities List Pointer (CSXE_SPI_CCAPP) — Offset 40h 2568
27.16 MSI Message Control, Next Pointer and Capability ID (CSXE_SPI_MSIMC_MSINP_MSICID)

— Offset 44h 2568
27.17 MSI Message Address (CSXE_SPI_MSIMA) — Offset 48h 2569
27.18 MSI Message Data (CSXE_SPI_MSIMD) — Offset 4Ch 2569
27.19 PCI Power Management Capability (CSXE_SPI_PMCAP_PMNP_PMCID) — Offset 50h 2570
27.20 PCI Power Management Control and Status (CSXE_SPI_PMD_PMCSRBSE_PMCSR) — Offset 54h 2570
27.21 CSXE Flash Control (CSXE_SPI_CSXEFCTRL) — Offset 80h 2571
27.22 CSXE Hardware Decompression Flash Limit (CSXE_SPI_CSXEHDFL) — Offset 84h ... 2572
27.23 SPI Unsupported Request Status (BIOS_SPI_UR_STS_CTL) — Offset D0h 2572
27.24 B-Unit Copy of the BIOS Decode Enable register for SPI (B_CR_BDE_SPI) — Offset D8h . 2573
27.25 BIOS Control (BIOS_SPI_BC) — Offset DCh 2573
27.26 Manufacturer's ID (BIOS_SPI_MANID) — Offset F8h 2575
27.27 SSP Control Register 0 (SSCR0)—Offset 0h 2575
27.28 SSP Control Register 1 (SSCR1)—Offset 4h 2576
27.29 SSP Status Register (SSSR)—Offset 8h 2578
27.30 SSP Interrupt Test Register (SSITR)—Offset Ch 2579
27.31 SSP Data (SSDR)—Offset 10h 2580
27.32 SSP Time Out (SSTO)—Offset 28h 2580
27.33 reserved (SSPSP_res)—Offset 2Ch 2581
27.34 reg_SSTSA_res (SSTSA_res)—Offset 30h 2581
27.35 SSRSA_Res (SSRSA_res)—Offset 34h 2581
27.36 SSTSS_res (SSTSS_res)—Offset 38h 2582
27.37 (SSACD_res)—Offset 3Ch 2582
27.38 Reserved (ITF_res)—Offset 40h 2582
27.39 SPI Transmit FIFO (SITF)—Offset 44h 2582
27.40 SPI Receive FIFO (SIRF)—Offset 48h 2583
27.41 TPM Reg 0 (TPM_REG_0)—Offset FED40000h 2583
27.42 TPM Reg 1 (TPM_REG_1)—Offset FED40004h 2584
27.43 TPM Reg 2 (TPM_REG_2)—Offset FED40008h 2584
27.44 TPM Reg 3 (TPM_REG_3)—Offset FED4000Ch 2584
27.45 TPM Reg 4 (TPM_REG_4)—Offset FED40010h 2585
27.46 CSXE Flash Primary Region (CSXE_MEFPREG)—Offset 0h 2585
27.47 Hardware Sequencing Flash Status and Control (CSXE_HSFSTS_CTL)—Offset 4h 2585
27.48 Flash Address (CSXE_FADDR)—Offset 8h 2588
27.49 Discrete Lock Bits (CSXE_DLOCK)—Offset Ch 2589
27.50 Flash Data 0 (CSXE_FDATA0)—Offset 10h 2589
27.51 Flash Data 1 (CSXE_FDATA1)—Offset 14h 2590
27.52 Flash Data 2 (CSXE_FDATA2)—Offset 18h 2590
27.53 Flash Data 3 (CSXE_FDATA3)—Offset 1Ch 2590
27.54 Flash Data 4 (CSXE_FDATA4)—Offset 20h 2591
27.55 Flash Data 5 (CSXE_FDATA5)—Offset 24h 2591
27.56 Flash Data 6 (CSXE_FDATA6)—Offset 28h 2591
27.57 Flash Data 7 (CSXE_FDATA7)—Offset 2Ch 2592
27.58 Flash Data 8 (CSXE_FDATA8)—Offset 30h 2592
27.59 Flash Data 9 (CSXE_FDATA9)—Offset 34h 2592
27.60 Flash Data 10 (CSXE_FDATA10)—Offset 38h 2593
27.61 Flash Data 11 (CSXE_FDATA11)—Offset 3Ch 2593
27.62 Flash Data 12 (CSXE_FDATA12)—Offset 40h 2593
27.63 Flash Data 13 (CSXE_FDATA13)—Offset 44h 2593
27.64 Flash Data 14 (CSXE_FDATA14)—Offset 48h 2594
27.65 Flash Data 15 (CSXE_FDATA15)—Offset 4Ch 2594
27.66 Flash Region Access Permissions (CSXE_FRACC)—Offset 50h 2594
27.67 Flash Region 0 (CSXE_FREG0)—Offset 54h 2595
27.68 Flash Region 1 (CSXE_FREG1)—Offset 58h 2596

27.69 Flash Region 2 (CSXE_FREG2)—Offset 5Ch 2596
27.70 Flash Region 3 (CSXE_FREG3)—Offset 60h..... 2596
27.71 Flash Region 4 (CSXE_FREG4)—Offset 64h..... 2597
27.72 Flash Region 5 (CSXE_FREG5)—Offset 68h..... 2597
27.73 Flash Region 6 (CSXE_FREG6)—Offset 6Ch 2598
27.74 Flash Region 7 (CSXE_FREG7)—Offset 70h..... 2598
27.75 Flash Region 8 (CSXE_FREG8)—Offset 74h..... 2599
27.76 Flash Region 9 (CSXE_FREG9)—Offset 78h..... 2599
27.77 Flash Region 10 (CSXE_FREG10)—Offset 7Ch 2600
27.78 Flash Region 11 (CSXE_FREG11)—Offset 80h 2600
27.79 Flash Protected Range 0 (CSXE_FPR0)—Offset 84h..... 2601
27.80 Flash Protected Range 1 (CSXE_FPR1)—Offset 88h..... 2601
27.81 Flash Protected Range 2 (CSXE_FPR2)—Offset 8Ch 2602
27.82 Flash Protected Range 3 (CSXE_FPR3)—Offset 90h..... 2603
27.83 Flash Protected Range 4 (CSXE_FPR4)—Offset 94h..... 2603
27.84 Write Protected Range 0 (CSXE_WPR0)—Offset 98h 2604
27.85 Software Sequencing Flash Status and Control (CSXE_SSFSTS_CTL)—Offset A0h ... 2605
27.86 Prefix Opcode and Opcode Type Configuration (CSXE_PREOP_OPTYPE)—Offset A4h 2608
27.87 Opcode Menu0 Configuration (CSXE_OPMENU0)—Offset A8h 2609
27.88 Opcode Menu1 Configuration (CSXE_OPMENU1)—Offset ACh 2610
27.89 Vendor Specific Component Capabilities for Component 0 (CSXE_SFDP0_VSCC0)—Offset C4h 2610
27.90 Vendor Specific Component Capabilities for Component 1 (CSXE_SFDP1_VSCC1)—Offset C8h 2611
27.91 Parameter Table Index (CSXE_PTINX)—Offset CCh..... 2612
27.92 Parameter Table Data (CSXE_PTDATA)—Offset D0h 2613
27.93 SPI Bus Requester Status (CSXE_SBRS)—Offset D4h..... 2613
27.94 Huffman Decompression Compressed Page Offset (CSXE_HDCOMPOFF)—Offset D8h 2614
27.95 Touch Controller Configuration Register (CSXE_TCCR)—Offset DCh 2615
27.96 Flash Region 12 (CSXE_FREG12)—Offset E0h 2615
27.97 Flash Region 13 (CSXE_FREG13)—Offset E4h 2616
27.98 Flash Region 14 (CSXE_FREG14)—Offset E8h 2616
27.99 Flash Region 15 (CSXE_FREG15)—Offset ECh 2617
27.100Interrupt Cause Enable (CSXE_ICE)—Offset F0h..... 2617
27.101Interrupt Cause Status (CSXE_ICS)—Offset F4h 2618
27.102Device 0 Write/Erase Count (CSXE_D0WEC)—Offset FCh 2619
27.103Device 1 Write/Erase Count (CSXE_D1WEC)—Offset 100h 2619
27.104Write/Erase Count Threshold (CSXE_WECTHR)—Offset 104h..... 2620
27.105RPMC SFDP Table (CSXE_RPMC0_D0)—Offset 108h 2620
27.106RPMC SFDP Table (CSXE_RPMC1_D0)—Offset 10Ch 2621
27.107RPMC SFDP Table (CSXE_RPMC0_D1)—Offset 110h 2621
27.108RPMC SFDP Table (CSXE_RPMC1_D1)—Offset 114h 2622
27.109CSME Master Read Access Permissions (CSXE_CM_RAP)—Offset 118h..... 2622
27.110CSME Master Write Access Permissions (CSXE_CM_WAP)—Offset 11Ch 2622
27.111Host Flash Protected Range 0 (CSXE_HOST_PR0)—Offset 184h 2623
27.112Host Flash Protected Range 1 (CSXE_HOST_PR1)—Offset 188h 2623
27.113Host Flash Protected Range 2 (CSXE_HOST_PR2)—Offset 18Ch 2624
27.114Host Flash Protected Range 3 (CSXE_HOST_PR3)—Offset 190h 2624
27.115Host Flash Protected Range 4 (CSXE_HOST_PR4)—Offset 194h 2625
27.116Host Global Protected Range 0 (CSXE_HOST_GPR0)—Offset 198h 2625
27.117IE Flash Protected Range 0 (CSXE_IE_PR0)—Offset 1B0h 2626
27.118IE Flash Protected Range 1 (CSXE_IE_PR1)—Offset 1B4h 2626
27.119IE Flash Protected Range 2 (CSXE_IE_PR2)—Offset 1B8h 2627
27.120IE Flash Protected Range 3 (CSXE_IE_PR3)—Offset 1BCh 2627

27.121IE Flash Protected Range 4 (CSXE_IE_PR4)—Offset 1C0h 2628
27.122BIOS Flash Primary Region (BIOS_BFPREG)—Offset 0h 2628
27.123Hardware Sequencing Flash Status and Control (BIOS_HSFSTS_CTL)—Offset 4h 2629
27.124Flash Address (BIOS_FADDR)—Offset 8h 2632
27.125Discrete Lock Bits (BIOS_DLOCK)—Offset Ch 2632
27.126Flash Data 0 (BIOS_FDATA0)—Offset 10h 2633
27.127Flash Data 1 (BIOS_FDATA1)—Offset 14h 2633
27.128Flash Data 2 (BIOS_FDATA2)—Offset 18h 2634
27.129Flash Data 3 (BIOS_FDATA3)—Offset 1Ch 2634
27.130Flash Data 4 (BIOS_FDATA4)—Offset 20h 2634
27.131Flash Data 5 (BIOS_FDATA5)—Offset 24h 2635
27.132Flash Data 6 (BIOS_FDATA6)—Offset 28h 2635
27.133Flash Data 7 (BIOS_FDATA7)—Offset 2Ch 2635
27.134Flash Data 8 (BIOS_FDATA8)—Offset 30h 2635
27.135Flash Data 9 (BIOS_FDATA9)—Offset 34h 2636
27.136Flash Data 10 (BIOS_FDATA10)—Offset 38h 2636
27.137Flash Data 11 (BIOS_FDATA11)—Offset 3Ch 2636
27.138Flash Data 12 (BIOS_FDATA12)—Offset 40h 2637
27.139Flash Data 13 (BIOS_FDATA13)—Offset 44h 2637
27.140Flash Data 14 (BIOS_FDATA14)—Offset 48h 2637
27.141Flash Data 15 (BIOS_FDATA15)—Offset 4Ch 2638
27.142Flash Region Access Permissions (BIOS_FRACC)—Offset 50h 2638
27.143Flash Region 0 (BIOS_FREG0)—Offset 54h 2639
27.144Flash Region 1 (BIOS_FREG1)—Offset 58h 2639
27.145Flash Region 2 (BIOS_FREG2)—Offset 5Ch 2640
27.146Flash Region 3 (BIOS_FREG3)—Offset 60h 2640
27.147Flash Region 4 (BIOS_FREG4)—Offset 64h 2641
27.148Flash Region 5 (BIOS_FREG5)—Offset 68h 2641
27.149Flash Region 6 (BIOS_FREG6)—Offset 6Ch 2642
27.150Flash Region 7 (BIOS_FREG7)—Offset 70h 2642
27.151Flash Region 8 (BIOS_FREG8)—Offset 74h 2642
27.152Flash Region 9 (BIOS_FREG9)—Offset 78h 2643
27.153Flash Region 10 (BIOS_FREG10)—Offset 7Ch 2643
27.154Flash Region 11 (BIOS_FREG11)—Offset 80h 2644
27.155Flash Protected Range 0 (BIOS_FPR0)—Offset 84h 2644
27.156Flash Protected Range 1 (BIOS_FPR1)—Offset 88h 2645
27.157Flash Protected Range 2 (BIOS_FPR2)—Offset 8Ch 2646
27.158Flash Protected Range 3 (BIOS_FPR3)—Offset 90h 2647
27.159Flash Protected Range 4 (BIOS_FPR4)—Offset 94h 2647
27.160Global Protected Range 0 (BIOS_GPR0)—Offset 98h 2648
27.161Software Sequencing Flash Status and Control (BIOS_SSFSTS_CTL)—Offset A0h 2649
27.162Prefix Opcode and Opcode Type Configuration (BIOS_PREOP_OPTYPE)—Offset A4h 2652
27.163Opcode Menu0 Configuration (BIOS_OPMENU0)—Offset A8h 2653
27.164Opcode Menu1 Configuration (BIOS_OPMENU1)—Offset ACh 2654
27.165Secondary Flash Region Access Permissions (BIOS_SFRACC)—Offset B0h 2654
27.166Flash Descriptor Observability Control (BIOS_FDOC)—Offset B4h 2655
27.167Flash Descriptor Observability Data (BIOS_FDOD)—Offset B8h 2655
27.168Additional Flash Control (BIOS_AFC)—Offset C0h 2656
27.169Vendor Specific Component Capabilities for Component 0 (BIOS_SFDP0_VSCC0)—Offset C4h 2656
27.170Vendor Specific Component Capabilities for Component 1 (BIOS_SFDP1_VSCC1)—Offset C8h 2659
27.171Parameter Table Index (BIOS_PTINX)—Offset CCh 2660
27.172Parameter Table Data (BIOS_PTDATA)—Offset D0h 2661
27.173SPI Bus Requester Status (BIOS_SBRS)—Offset D4h 2661

27.174Flash Region 12 (BIOS_FREG12)—Offset E0h 2662
27.175Flash Region 13 (BIOS_FREG13)—Offset E4h 2663
27.176Flash Region 14 (BIOS_FREG14)—Offset E8h 2663
27.177Flash Region 15 (BIOS_FREG15)—Offset ECh 2664
27.178RPMC SFDP Table (BIOS_RPMC0_D0)—Offset 108h 2664
27.179RPMC SFDP Table (BIOS_RPMC1_D0)—Offset 10Ch 2664
27.180RPMC SFDP Table (BIOS_RPMC0_D1)—Offset 110h 2665
27.181RPMC SFDP Table (BIOS_RPMC1_D1)—Offset 114h 2665
27.182BIOS Master Read Access Permissions (BIOS_BM_RAP)—Offset 118h 2665
27.183BIOS Master Write Access Permissions (BIOS_BM_WAP)—Offset 11Ch 2666
27.184CSXE Flash Protected Range 0 (BIOS_CSXE_PR0)—Offset 184h 2666
27.185CSXE Flash Protected Range 1 (BIOS_CSXE_PR1)—Offset 188h 2667
27.186CSXE Flash Protected Range 2 (BIOS_CSXE_PR2)—Offset 18Ch 2667
27.187CSXE Flash Protected Range 3 (BIOS_CSXE_PR3)—Offset 190h 2668
27.188CSXE Flash Protected Range 4 (BIOS_CSXE_PR4)—Offset 194h 2668
27.189Write Protected Range 0 (BIOS_CSXE_WPR0)—Offset 198h 2669
27.190IE Flash Protected Range 0 (BIOS_IE_PR0)—Offset 1B0h 2670
27.191IE Flash Protected Range 1 (BIOS_IE_PR1)—Offset 1B4h 2670
27.192IE Flash Protected Range 2 (BIOS_IE_PR2)—Offset 1B8h 2671
27.193IE Flash Protected Range 3 (BIOS_IE_PR3)—Offset 1BCh 2671
27.194IE Flash Protected Range 4 (BIOS_IE_PR4)—Offset 1C0h 2672
27.195eSPI Controller HW Status 1 (ESPI_HW_STS_2)—Offset 18h 2672
27.196 (EC_FLASH_WR_DATA_0)—Offset 1A0h 2673
27.197 (EC_FLASH_WR_DATA_1)—Offset 1A4h 2673
27.198 (EC_FLASH_WR_DATA_2)—Offset 1A8h 2674
27.199 (EC_FLASH_WR_DATA_3)—Offset 1ACh 2674
27.200 (EC_FLASH_WR_DATA_4)—Offset 1B0h 2674
27.201 (EC_FLASH_WR_DATA_5)—Offset 1B4h 2675
27.202 (EC_FLASH_WR_DATA_6)—Offset 1B8h 2675
27.203 (EC_FLASH_WR_DATA_7)—Offset 1BCh 2675
27.204 (EC_FLASH_WR_DATA_8)—Offset 1C0h 2675
27.205 (EC_FLASH_WR_DATA_9)—Offset 1C4h 2676
27.206 (EC_FLASH_WR_DATA_10)—Offset 1C8h 2676
27.207 (EC_FLASH_WR_DATA_11)—Offset 1CCh 2676
27.208 (EC_FLASH_WR_DATA_12)—Offset 1D0h 2676
27.209 (EC_FLASH_WR_DATA_13)—Offset 1D4h 2677
27.210 (EC_FLASH_WR_DATA_14)—Offset 1D8h 2677
27.211 (EC_FLASH_WR_DATA_15)—Offset 1DCh 2677
27.212 (EC_FLASH_REQ_ADDR)—Offset 1E0h 2678
27.213 (EC_FLASH_CPL_CTL)—Offset 1E4h 2678
27.214(EC_FLASH_CPL_STS)—Offset 1E8h 2678
27.215 (EC_FLASH_RD_DATA_0)—Offset 200h 2679
27.216 (EC_FLASH_RD_DATA_1)—Offset 204h 2679
27.217 (EC_FLASH_RD_DATA_2)—Offset 208h 2679
27.218 (EC_FLASH_RD_DATA_3)—Offset 20Ch 2680
27.219 (EC_FLASH_RD_DATA_4)—Offset 210h 2680
27.220 (EC_FLASH_RD_DATA_5)—Offset 214h 2680
27.221 (EC_FLASH_RD_DATA_6)—Offset 218h 2680
27.222 (EC_FLASH_RD_DATA_7)—Offset 21Ch 2681
27.223 (EC_FLASH_RD_DATA_8)—Offset 220h 2681
27.224 (EC_FLASH_RD_DATA_9)—Offset 224h 2681
27.225 (EC_FLASH_RD_DATA_10)—Offset 228h 2681
27.226 (EC_FLASH_RD_DATA_11)—Offset 22Ch 2682
27.227 (EC_FLASH_RD_DATA_12)—Offset 230h 2682
27.228 (EC_FLASH_RD_DATA_13)—Offset 234h 2682

27.229 (EC_FLASH_RD_DATA_14)—Offset 238h....2683
27.230 (EC_FLASH_RD_DATA_15)—Offset 23Ch....2683
27.231Device and Vendor ID (CSXE_ESPI_DID_VID)—Offset 0h....2683
27.232Device Control (CSXE_ESPI_STS_CMD)—Offset 4h....2683
27.233Revision ID and Class Code (CSXE_ESPI_CC_RID)—Offset 8h....2685
27.234CSXE BAR0 MMIO (CSXE_ESPI_MBAR)—Offset 10h....2686
27.235Capabilities List Pointer (CSXE_ESPI_CAPP)—Offset 40h....2686
27.236MSI Message Control, Next Pointer and Capability ID (CSXE_ESPI_MSIMC_MSINP_MSICID)—Offset 44h....2687
27.237MSI Message Address (CSXE_ESPI_MSIMA)—Offset 48h....2687
27.238MSI Message Data (CSXE_ESPI_MSIMD)—Offset 4Ch....2688
27.239PCI Power Management Capability (CSXE_ESPI_PMCAP_PMNP_PMCID)—Offset 50h2688
27.240PCI Power Management Control and Status (CSXE_ESPI_PMD_PMCSRBSE_PMCSR)—Offset 54h....2689
27.241eSPI MSI cause (CSXE_ESPI_MSI_STS)—Offset 0h....2691
27.242eSPI MSI Enables (CSXE_ESPI_MSI_EN)—Offset 4h....2692
27.243eSPI Controller HW Status 0 (CSXE_ESPI_HW_STS_0)—Offset 10h....2693
27.244eSPI Controller HW Status 1 (CSXE_ESPI_HW_STS_1)—Offset 14h....2694
27.245eSPI OOB Channel Controller and Status 0 (CSXE_ESPI_OOB_CTL_STS_0)—Offset 1Ch.2695
27.246eSPI OOB Channel Controller and Status 1 (CSXE_ESPI_OOB_CTL_STS_1)—Offset 20h.2697
27.247OOB SMBus Tx Control (CSXE_ESPI_OOB_SMBUS_TX_CTL)—Offset 2Ch....2698
27.248 (CSXE_ESPI_OOB_SMBUS_TX_DATA_0)—Offset 30h....2699
27.249 (CSXE_ESPI_OOB_SMBUS_TX_DATA_1)—Offset 34h....2700
27.250 (CSXE_ESPI_OOB_SMBUS_TX_DATA_2)—Offset 38h....2700
27.251 (CSXE_ESPI_OOB_SMBUS_TX_DATA_3)—Offset 3Ch....2700
27.252 (CSXE_ESPI_OOB_SMBUS_TX_DATA_4)—Offset 40h....2701
27.253 (CSXE_ESPI_OOB_SMBUS_TX_DATA_5)—Offset 44h....2701
27.254 (CSXE_ESPI_OOB_SMBUS_TX_DATA_6)—Offset 48h....2701
27.255 (CSXE_ESPI_OOB_SMBUS_TX_DATA_7)—Offset 4Ch....2701
27.256 (CSXE_ESPI_OOB_SMBUS_TX_DATA_8)—Offset 50h....2702
27.257 (CSXE_ESPI_OOB_SMBUS_TX_DATA_9)—Offset 54h....2702
27.258 (CSXE_ESPI_OOB_SMBUS_TX_DATA_10)—Offset 58h....2702
27.259 (CSXE_ESPI_OOB_SMBUS_TX_DATA_11)—Offset 5Ch....2703
27.260 (CSXE_ESPI_OOB_SMBUS_TX_DATA_12)—Offset 60h....2703
27.261 (CSXE_ESPI_OOB_SMBUS_TX_DATA_13)—Offset 64h....2703
27.262 (CSXE_ESPI_OOB_SMBUS_TX_DATA_14)—Offset 68h....2703
27.263 (CSXE_ESPI_OOB_SMBUS_TX_DATA_15)—Offset 6Ch....2704
27.264 (CSXE_ESPI_OOB_SMBUS_TX_DATA_16)—Offset 70h....2704
27.265 (CSXE_ESPI_OOB_SMBUS_TX_DATA_17)—Offset 74h....2704
27.266 (CSXE_ESPI_OOB_SMBUS_TX_DATA_18)—Offset 78h....2705
27.267 (CSXE_ESPI_OOB_SMBUS_RX_DATA_0)—Offset 150h....2705
27.268 (CSXE_ESPI_OOB_SMBUS_RX_DATA_1)—Offset 154h....2705
27.269(CSXE_ESPI_OOB_SMBUS_RX_DATA_2)—Offset 158h....2706
27.270(CSXE_ESPI_OOB_SMBUS_RX_DATA_3)—Offset 15Ch....2706
27.271(CSXE_ESPI_OOB_SMBUS_RX_DATA_4)—Offset 160h....2706
27.272(CSXE_ESPI_OOB_SMBUS_RX_DATA_5)—Offset 164h....2706
27.273(CSXE_ESPI_OOB_SMBUS_RX_DATA_6)—Offset 168h....2707
27.274(CSXE_ESPI_OOB_SMBUS_RX_DATA_7)—Offset 16Ch....2707
27.275(CSXE_ESPI_OOB_SMBUS_RX_DATA_8)—Offset 170h....2707
27.276(CSXE_ESPI_OOB_SMBUS_RX_DATA_9)—Offset 174h....2708
27.277(CSXE_ESPI_OOB_SMBUS_RX_DATA_10)—Offset 178h....2708
27.278(CSXE_ESPI_OOB_SMBUS_RX_DATA_11)—Offset 17Ch....2708
27.279(CSXE_ESPI_OOB_SMBUS_RX_DATA_12)—Offset 180h....2708

27.280(CSXE_ESPI_OOB_SMBUS_RX_DATA_13)—Offset 184h 2709
27.281(CSXE_ESPI_OOB_SMBUS_RX_DATA_14)—Offset 188h 2709
27.282(CSXE_ESPI_OOB_SMBUS_RX_DATA_15)—Offset 18Ch 2709
27.283(CSXE_ESPI_OOB_SMBUS_RX_DATA_16)—Offset 190h 2709
27.284(CSXE_ESPI_OOB_SMBUS_RX_DATA_17)—Offset 194h 2710
27.285(CSXE_ESPI_OOB_SMBUS_RX_DATA_18)—Offset 198h 2710

28 CNVi Registers.....2711

28.1 CNVI_WIFI_VEN_DEV_ID — Offset 0h 2711
28.2 CNVI_WIFI_PCI_COM_STAT — Offset 4h..... 2711
28.3 CNVI_WIFI_PCI_CLASS_CODE — Offset 8h 2712
28.4 CNVI_WIFI_PCI_BUS_COM — Offset Ch..... 2713
28.5 CNVI_WIFI_BAR0 — Offset 10h 2713
28.6 CNVI_WIFI_BAR1 — Offset 14h 2714
28.7 CNVI_WIFI_BAR2 — Offset 18h 2714
28.8 CNVI_WIFI_BAR3 — Offset 1Ch 2714
28.9 CNVI_WIFI_BAR4 — Offset 20h 2715
28.10 CNVI_WIFI_BAR5 — Offset 24h 2715
28.11 CNVI_WIFI_CIS_PTR — Offset 28h..... 2715
28.12 CNVI_WIFI_SUBSYS_ID — Offset 2Ch 2716
28.13 CNVI_WIFI_ROM_BAR — Offset 30h..... 2716
28.14 CNVI_WIFI_CAP_PTR — Offset 34h 2716
28.15 CNVI_WIFI_RES_ADDRESS — Offset 38h..... 2717
28.16 CNVI_WIFI_INTERRUPT — Offset 3Ch..... 2717
28.17 CNVI_WIFI_GIO_CAP — Offset 40h..... 2717
28.18 CNVI_WIFI_GIO_DEV_CAP — Offset 44h 2718
28.19 CNVI_WIFI_GIO_DEV — Offset 48h..... 2719
28.20 CNVI_WIFI_GIO_LNK_CAP — Offset 4Ch 2720
28.21 CNVI_WIFI_GIO_LINK — Offset 50h..... 2720
28.22 CNVI_WIFI_GIO_DEV_CAP_2 — Offset 64h..... 2720
28.23 CNVI_WIFI_GIO_DEV_2 — Offset 68h 2721
28.24 CNVI_WIFI_GIO_LINK_2 — Offset 70h 2722
28.25 CNVI_WIFI_MSIX_CAP_HEAD — Offset 80h..... 2722
28.26 CNVI_WIFI_MSIX_TABLE_OFFSET — Offset 84h 2722
28.27 CNVI_WIFI_MSIX_PBA_OFFSET — Offset 88h..... 2723
28.28 CNVI_WIFI_PMC — Offset C8h 2723
28.29 CNVI_WIFI_PMCSR — Offset CCh..... 2724
28.30 CNVI_WIFI_MSI_MSG_CTRL — Offset D0h 2725
28.31 CNVI_WIFI_MSI_LOW_ADD — Offset D4h..... 2725
28.32 CNVI_WIFI_MSI_HIGH_ADD — Offset D8h..... 2725
28.33 CNVI_WIFI_MSI_DATA — Offset DCh 2726
28.34 CNVI_WIFI_GIO_SERIAL_CAP — Offset 100h 2726
28.35 CNVI_WIFI_GIO_SERIAL_LOW — Offset 104h 2726
28.36 CNVI_WIFI_GIO_SERIAL_UP — Offset 108h 2727
28.37 CNVI_WIFI_UNCORRECT_ERR_SEV — Offset 10Ch..... 2727
28.38 CNVI_WIFI_CORRECT_ERR_STAT — Offset 110h 2728
28.39 CNVI_WIFI_CORRECT_ERR_MASK — Offset 114h..... 2729
28.40 CNVI_WIFI_ADVANCED_ERR_CAP — Offset 118h..... 2730
28.41 CNVI_WIFI_HEADER_LOG1 — Offset 11Ch..... 2730
28.42 CNVI_WIFI_HEADER_LOG2 — Offset 120h..... 2731
28.43 CNVI_WIFI_HEADER_LOG3 — Offset 124h..... 2731
28.44 CNVI_WIFI_HEADER_LOG4 — Offset 128h..... 2731
28.45 CNVI_WIFI_LTR_EXTND_CAP_HEAD — Offset 14Ch 2731
28.46 CNVI_WIFI_LTR_MAX_SNOOP_NOSNOOP_LAT — Offset 150h 2732
28.47 CNVI_WIFI_L1PM_SUB_EXTND_CAP_HEAD — Offset 154h..... 2732

28.48 CNVI_WIFI_L1PM_SUB_CAP — Offset 158h 2733
28.49 CNVI_WIFI_L1PM_SUB_CNTRL — Offset 15Ch....2733
28.50 CNVI_WIFI_L1PM_SUB_CNTRL2 — Offset 160h 2734
28.51 CNVI_WIFI_VEN_SPEC_CAP — Offset 164h....2734
28.52 CNVI_WIFI_VEN_SPEC_EXTND_CAP — Offset 168h 2735
28.53 CNVI_WIFI_LTP_PTR — Offset 16Ch 2735
28.54 CNVI_WIFI_DEV_IDLE_PTR — Offset 170h 2736
28.55 CNVI_WIFI_DEV_IDLE_PWR — Offset 174h 2736

29 GPIO Registers 2738

29.1 Revision ID (REV_ID)—Offset 0h 2738
29.2 Capability List Register (CAP_LIST_0)—Offset 4h....2738
29.3 Family Base Address (FAMBAR)—Offset 8h 2738
29.4 Pad Base Address (PADBAR)—Offset Ch 2739
29.5 Miscellaneous Configuration (MISCCFG)—Offset 10h 2739
29.6 Miscellaneous Secured Configuration (MISCSECCFG)—Offset 14h....2740
29.7 Reserved (RSVD0[0])—Offset 18h 2741
29.8 Reserved (RSVD0[1])—Offset 1Ch 2741
29.9 Pad Ownership (PAD_OWN_audio_0)—Offset 20h 2741
29.10 Pad Ownership (PAD_OWN_audio_1)—Offset 24h 2746
29.11 Pad Ownership (PAD_OWN_audio_2)—Offset 28h 2750
29.12 Pad Ownership (PAD_OWN_audio_3)—Offset 2Ch 2755
29.13 Reserved (RSVD1[0])—Offset 30h 2757
29.14 Reserved (RSVD1[1])—Offset 34h 2757
29.15 Reserved (RSVD1[2])—Offset 38h 2758
29.16 Reserved (RSVD1[3])—Offset 3Ch 2758
29.17 Reserved (RSVD1[4])—Offset 40h 2758
29.18 Reserved (RSVD1[5])—Offset 44h 2758
29.19 Reserved (RSVD1[6])—Offset 48h 2759
29.20 Reserved (RSVD1[7])—Offset 4Ch 2759
29.21 Reserved (RSVD1[8])—Offset 50h 2759
29.22 Reserved (RSVD1[9])—Offset 54h 2760
29.23 Reserved (RSVD1[10])—Offset 58h 2760
29.24 Reserved (RSVD1[11])—Offset 5Ch 2760
29.25 GPI Virtual Wire Message Enable (GPI_VWE_audio_0)—Offset 60h....2760
29.26 Reserved (RSVD2[0])—Offset 64h 2768
29.27 Reserved (RSVD2[1])—Offset 68h 2768
29.28 Reserved (RSVD2[2])—Offset 6Ch 2768
29.29 Reserved (RSVD2[3])—Offset 70h 2769
29.30 Reserved (RSVD2[4])—Offset 74h 2769
29.31 Reserved (RSVD2[5])—Offset 78h 2769
29.32 Reserved (RSVD2[6])—Offset 7Ch 2769
29.33 Pad Configuration Lock (PADCFGLOCK_audio_0)—Offset 80h 2770
29.34 Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_audio_0)—Offset 84h... 2784
29.35 Reserved (RSVD3[0])—Offset 88h 2798
29.36 Reserved (RSVD3[1])—Offset 8Ch 2799
29.37 Reserved (RSVD3[2])—Offset 90h 2799
29.38 Reserved (RSVD3[3])—Offset 94h 2799
29.39 Reserved (RSVD3[4])—Offset 98h 2799
29.40 Reserved (RSVD3[5])—Offset 9Ch 2800
29.41 Reserved (RSVD3[6])—Offset A0h 2800
29.42 Reserved (RSVD3[7])—Offset A4h 2800
29.43 Reserved (RSVD3[8])—Offset A8h 2800
29.44 Reserved (RSVD3[9])—Offset ACh 2801

29.45 Host Software Pad Ownership (HOSTSW_OWN_audio_0)—Offset B0h 2801
29.46 Reserved (RSVD4[0])—Offset B4h 2807
29.47 Reserved (RSVD4[1])—Offset B8h 2807
29.48 Reserved (RSVD4[2])—Offset BCh 2807
29.49 Reserved (RSVD4[3])—Offset C0h 2808
29.50 Reserved (RSVD4[4])—Offset C4h 2808
29.51 Reserved (RSVD4[5])—Offset C8h 2808
29.52 Reserved (RSVD4[6])—Offset CCh 2809
29.53 Reserved (RSVD4[7])—Offset D0h 2809
29.54 Reserved (RSVD4[8])—Offset D4h 2809
29.55 Reserved (RSVD4[9])—Offset D8h 2809
29.56 Reserved (RSVD4[10])—Offset DCh 2810
29.57 Reserved (RSVD4[11])—Offset E0h 2810
29.58 Reserved (RSVD4[12])—Offset E4h 2810
29.59 Reserved (RSVD4[13])—Offset E8h 2810
29.60 Reserved (RSVD4[14])—Offset ECh 2811
29.61 Reserved (RSVD4[15])—Offset F0h 2811
29.62 Reserved (RSVD4[16])—Offset F4h 2811
29.63 Reserved (RSVD4[17])—Offset F8h 2812
29.64 Reserved (RSVD4[18])—Offset FCh 2812
29.65 GPI Interrupt Status (GPI_IS_audio_0)—Offset 100h 2812
29.66 Reserved (RSVD5[0])—Offset 104h 2819
29.67 Reserved (RSVD5[1])—Offset 108h 2820
29.68 Reserved (RSVD5[2])—Offset 10Ch 2820
29.69 GPI Interrupt Enable (GPI_IE_audio_0)—Offset 110h 2820
29.70 Reserved (RSVD6[0])—Offset 114h 2825
29.71 Reserved (RSVD6[1])—Offset 118h 2825
29.72 Reserved (RSVD6[2])—Offset 11Ch 2826
29.73 Reserved (RSVD6[3])—Offset 120h 2826
29.74 Reserved (RSVD6[4])—Offset 124h 2826
29.75 Reserved (RSVD6[5])—Offset 128h 2826
29.76 Reserved (RSVD6[6])—Offset 12Ch 2827
29.77 GPI General Purpose Events Status (GPI_GPE_STS_audio_0)—Offset 130h 2827
29.78 Reserved (RSVD7[0])—Offset 134h 2834
29.79 Reserved (RSVD7[1])—Offset 138h 2835
29.80 Reserved (RSVD7[2])—Offset 13Ch 2835
29.81 Reserved (RSVD7[3])—Offset 140h 2835
29.82 Reserved (RSVD7[4])—Offset 144h 2835
29.83 Reserved (RSVD7[5])—Offset 148h 2836
29.84 Reserved (RSVD7[6])—Offset 14Ch 2836
29.85 GPI General Purpose Events Enable (GPI_GPE_EN_audio_0)—Offset 150h 2836
29.86 Reserved (RSVD8[0])—Offset 154h 2842
29.87 Reserved (RSVD8[1])—Offset 158h 2842
29.88 Reserved (RSVD8[2])—Offset 15Ch 2843
29.89 Reserved (RSVD8[3])—Offset 160h 2843
29.90 Reserved (RSVD8[4])—Offset 164h 2843
29.91 Reserved (RSVD8[5])—Offset 168h 2844
29.92 Reserved (RSVD8[6])—Offset 16Ch 2844
29.93 SMI Status (GPI_SMI_STS_audio_0)—Offset 170h 2844
29.94 Reserved (RSVD9[0])—Offset 174h 2854
29.95 Reserved (RSVD9[1])—Offset 178h 2854
29.96 Reserved (RSVD9[2])—Offset 17Ch 2854
29.97 Reserved (RSVD9[3])—Offset 180h 2855
29.98 Reserved (RSVD9[4])—Offset 184h 2855
29.99 Reserved (RSVD9[5])—Offset 188h 2855

29.100Reserved (RSVD9[6])—Offset 18Ch....2856
29.101SMI Enable (GPI_SMI_EN_audio_0)—Offset 190h....2856
29.102Reserved (RSVD10[0])—Offset 194h....2863
29.103Reserved (RSVD10[1])—Offset 198h....2863
29.104Reserved (RSVD10[2])—Offset 19Ch....2864
29.105Reserved (RSVD10[3])—Offset 1A0h....2864
29.106Reserved (RSVD10[4])—Offset 1A4h....2864
29.107Reserved (RSVD10[5])—Offset 1A8h....2864
29.108Reserved (RSVD10[6])—Offset 1ACh....2865
29.109NMI Status (GPI_NMI_STS_audio_0)—Offset 1B0h....2865
29.110Reserved (RSVD11[0])—Offset 1B4h....2872
29.111Reserved (RSVD11[1])—Offset 1B8h....2873
29.112Reserved (RSVD11[2])—Offset 1BCh....2873
29.113Reserved (RSVD11[3])—Offset 1C0h....2873
29.114Reserved (RSVD11[4])—Offset 1C4h....2873
29.115Reserved (RSVD11[5])—Offset 1C8h....2874
29.116Reserved (RSVD11[6])—Offset 1CCh....2874
29.117NMI Enable (GPI_NMI_EN_audio_0)—Offset 1D0h....2874
29.118Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_AVS_1P8)—Offset 300h....2881
29.119Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_AVS_1P8)—Offset 304h....2882
29.120Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_AVS_1P8)—Offset 308h....2882
29.121Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_AVS_1P8)—Offset 30Ch....2883
29.122Family Configuration Register (FAM_CFG_AVS_1P8)—Offset 310h....2883
29.123Family Reserved Register DW5 (FAM_RSVD_DW5AVS_1P8)—Offset 314h....2885
29.124Family Reserved Register DW6 (FAM_RSVD_DW6AVS_1P8)—Offset 318h....2885
29.125Family Reserved Register DW7 (FAM_RSVD_DW7AVS_1P8)—Offset 31Ch....2885
29.126Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_v_CNV1)—Offset 320h....2886
29.127Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_v_CNV1)—Offset 324h....2886
29.128Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_v_CNV1)—Offset 328h....2887
29.129Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_v_CNV1)—Offset 32Ch....2887
29.130Family Configuration Register (FAM_CFG_v_CNV1)—Offset 330h....2887
29.131Family Reserved Register DW5 (FAM_RSVD_DW5v_CNV1)—Offset 334h....2889
29.132Family Reserved Register DW6 (FAM_RSVD_DW6v_CNV1)—Offset 338h....2889
29.133Family Reserved Register DW7 (FAM_RSVD_DW7v_CNV1)—Offset 33Ch....2890
29.134Pad Configuration DW0 (PAD_CFG_DW0_GPIO_156)—Offset 600h....2890
29.135Pad Configuration DW1 (PAD_CFG_DW1_GPIO_156)—Offset 604h....2892
29.136Pad Configuration DW2 (PAD_CFG_DW2_GPIO_156)—Offset 608h....2894
29.137Pad Configuration DW3 (PAD_CFG_DW3_GPIO_156)—Offset 60Ch....2895
29.138Pad Configuration DW0 (PAD_CFG_DW0_GPIO_157)—Offset 610h....2895
29.139Pad Configuration DW1 (PAD_CFG_DW1_GPIO_157)—Offset 614h....2897
29.140Pad Configuration DW2 (PAD_CFG_DW2_GPIO_157)—Offset 618h....2899
29.141Pad Configuration DW3 (PAD_CFG_DW3_GPIO_157)—Offset 61Ch....2900
29.142Pad Configuration DW0 (PAD_CFG_DW0_GPIO_158)—Offset 620h....2900
29.143Pad Configuration DW1 (PAD_CFG_DW1_GPIO_158)—Offset 624h....2902
29.144Pad Configuration DW2 (PAD_CFG_DW2_GPIO_158)—Offset 628h....2904
29.145Pad Configuration DW3 (PAD_CFG_DW3_GPIO_158)—Offset 62Ch....2905
29.146Pad Configuration DW0 (PAD_CFG_DW0_GPIO_159)—Offset 630h....2905
29.147Pad Configuration DW1 (PAD_CFG_DW1_GPIO_159)—Offset 634h....2907
29.148Pad Configuration DW2 (PAD_CFG_DW2_GPIO_159)—Offset 638h....2909
29.149Pad Configuration DW3 (PAD_CFG_DW3_GPIO_159)—Offset 63Ch....2910
29.150Pad Configuration DW0 (PAD_CFG_DW0_GPIO_160)—Offset 640h....2910
29.151Pad Configuration DW1 (PAD_CFG_DW1_GPIO_160)—Offset 644h....2912
29.152Pad Configuration DW2 (PAD_CFG_DW2_GPIO_160)—Offset 648h....2914
29.153Pad Configuration DW3 (PAD_CFG_DW3_GPIO_160)—Offset 64Ch....2915
29.154Pad Configuration DW0 (PAD_CFG_DW0_GPIO_161)—Offset 650h....2915

29.155Pad Configuration DW1 (PAD_CFG_DW1_GPIO_161)—Offset 654h..... 2917
29.156Pad Configuration DW2 (PAD_CFG_DW2_GPIO_161)—Offset 658h..... 2919
29.157Pad Configuration DW3 (PAD_CFG_DW3_GPIO_161)—Offset 65Ch 2920
29.158Pad Configuration DW0 (PAD_CFG_DW0_GPIO_162)—Offset 660h..... 2920
29.159Pad Configuration DW1 (PAD_CFG_DW1_GPIO_162)—Offset 664h..... 2922
29.160Pad Configuration DW2 (PAD_CFG_DW2_GPIO_162)—Offset 668h..... 2924
29.161Pad Configuration DW3 (PAD_CFG_DW3_GPIO_162)—Offset 66Ch 2925
29.162Pad Configuration DW0 (PAD_CFG_DW0_GPIO_163)—Offset 670h..... 2925
29.163Pad Configuration DW1 (PAD_CFG_DW1_GPIO_163)—Offset 674h..... 2927
29.164Pad Configuration DW2 (PAD_CFG_DW2_GPIO_163)—Offset 678h..... 2929
29.165Pad Configuration DW3 (PAD_CFG_DW3_GPIO_163)—Offset 67Ch 2930
29.166Pad Configuration DW0 (PAD_CFG_DW0_GPIO_164)—Offset 680h..... 2930
29.167Pad Configuration DW1 (PAD_CFG_DW1_GPIO_164)—Offset 684h..... 2932
29.168Pad Configuration DW2 (PAD_CFG_DW2_GPIO_164)—Offset 688h..... 2934
29.169Pad Configuration DW3 (PAD_CFG_DW3_GPIO_164)—Offset 68Ch 2935
29.170Pad Configuration DW0 (PAD_CFG_DW0_GPIO_165)—Offset 690h..... 2935
29.171Pad Configuration DW1 (PAD_CFG_DW1_GPIO_165)—Offset 694h..... 2937
29.172Pad Configuration DW2 (PAD_CFG_DW2_GPIO_165)—Offset 698h..... 2939
29.173Pad Configuration DW3 (PAD_CFG_DW3_GPIO_165)—Offset 69Ch 2940
29.174Pad Configuration DW0 (PAD_CFG_DW0_GPIO_166)—Offset 6A0h 2940
29.175Pad Configuration DW1 (PAD_CFG_DW1_GPIO_166)—Offset 6A4h 2942
29.176Pad Configuration DW2 (PAD_CFG_DW2_GPIO_166)—Offset 6A8h 2944
29.177Pad Configuration DW3 (PAD_CFG_DW3_GPIO_166)—Offset 6ACh 2945
29.178Pad Configuration DW0 (PAD_CFG_DW0_GPIO_167)—Offset 6B0h 2945
29.179Pad Configuration DW1 (PAD_CFG_DW1_GPIO_167)—Offset 6B4h 2947
29.180Pad Configuration DW2 (PAD_CFG_DW2_GPIO_167)—Offset 6B8h 2949
29.181Pad Configuration DW3 (PAD_CFG_DW3_GPIO_167)—Offset 6BCh 2950
29.182Pad Configuration DW0 (PAD_CFG_DW0_GPIO_168)—Offset 6C0h 2950
29.183Pad Configuration DW1 (PAD_CFG_DW1_GPIO_168)—Offset 6C4h 2952
29.184Pad Configuration DW2 (PAD_CFG_DW2_GPIO_168)—Offset 6C8h 2954
29.185Pad Configuration DW3 (PAD_CFG_DW3_GPIO_168)—Offset 6CCh 2955
29.186Pad Configuration DW0 (PAD_CFG_DW0_GPIO_169)—Offset 6D0h 2955
29.187Pad Configuration DW1 (PAD_CFG_DW1_GPIO_169)—Offset 6D4h 2957
29.188Pad Configuration DW2 (PAD_CFG_DW2_GPIO_169)—Offset 6D8h 2959
29.189Pad Configuration DW3 (PAD_CFG_DW3_GPIO_169)—Offset 6DCh 2960
29.190Pad Configuration DW0 (PAD_CFG_DW0_GPIO_170)—Offset 6E0h..... 2960
29.191Pad Configuration DW1 (PAD_CFG_DW1_GPIO_170)—Offset 6E4h..... 2962
29.192Pad Configuration DW2 (PAD_CFG_DW2_GPIO_170)—Offset 6E8h..... 2964
29.193Pad Configuration DW3 (PAD_CFG_DW3_GPIO_170)—Offset 6ECh 2965
29.194Pad Configuration DW0 (PAD_CFG_DW0_GPIO_171)—Offset 6F0h..... 2965
29.195Pad Configuration DW1 (PAD_CFG_DW1_GPIO_171)—Offset 6F4h..... 2967
29.196Pad Configuration DW2 (PAD_CFG_DW2_GPIO_171)—Offset 6F8h..... 2969
29.197Pad Configuration DW3 (PAD_CFG_DW3_GPIO_171)—Offset 6FCh..... 2970
29.198Pad Configuration DW0 (PAD_CFG_DW0_GPIO_172)—Offset 700h..... 2970
29.199Pad Configuration DW1 (PAD_CFG_DW1_GPIO_172)—Offset 704h..... 2972
29.200Pad Configuration DW2 (PAD_CFG_DW2_GPIO_172)—Offset 708h..... 2974
29.201Pad Configuration DW3 (PAD_CFG_DW3_GPIO_172)—Offset 70Ch 2975
29.202Pad Configuration DW0 (PAD_CFG_DW0_GPIO_173)—Offset 710h..... 2975
29.203Pad Configuration DW1 (PAD_CFG_DW1_GPIO_173)—Offset 714h..... 2977
29.204Pad Configuration DW2 (PAD_CFG_DW2_GPIO_173)—Offset 718h..... 2979
29.205Pad Configuration DW3 (PAD_CFG_DW3_GPIO_173)—Offset 71Ch 2980
29.206Pad Configuration DW0 (PAD_CFG_DW0_GPIO_174)—Offset 720h..... 2980
29.207Pad Configuration DW1 (PAD_CFG_DW1_GPIO_174)—Offset 724h..... 2982
29.208Pad Configuration DW2 (PAD_CFG_DW2_GPIO_174)—Offset 728h..... 2984
29.209Pad Configuration DW3 (PAD_CFG_DW3_GPIO_174)—Offset 72Ch 2985

29.210Pad Configuration DW0 (PAD_CFG_DW0_GPIO_175)—Offset 730h 2985
29.211Pad Configuration DW1 (PAD_CFG_DW1_GPIO_175)—Offset 734h 2987
29.212Pad Configuration DW2 (PAD_CFG_DW2_GPIO_175)—Offset 738h 2989
29.213Pad Configuration DW3 (PAD_CFG_DW3_GPIO_175)—Offset 73Ch 2990
29.214Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_31)—Offset 740h 2990
29.215Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_31)—Offset 744h 2992
29.216Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_31)—Offset 748h 2994
29.217Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_31)—Offset 74Ch 2995
29.218Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_32)—Offset 750h 2995
29.219Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_32)—Offset 754h 2997
29.220Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_32)—Offset 758h 2999
29.221Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_32)—Offset 75Ch 3000
29.222Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_33)—Offset 760h 3000
29.223Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_33)—Offset 764h 3002
29.224Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_33)—Offset 768h 3004
29.225Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_33)—Offset 76Ch 3005
29.226Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_34)—Offset 770h 3005
29.227Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_34)—Offset 774h 3007
29.228Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_34)—Offset 778h 3009
29.229Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_34)—Offset 77Ch 3010
29.230Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_35)—Offset 780h 3010
29.231Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_35)—Offset 784h 3012
29.232Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_35)—Offset 788h 3014
29.233Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_35)—Offset 78Ch 3015
29.234Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_36)—Offset 790h 3015
29.235Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_36)—Offset 794h 3017
29.236Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_36)—Offset 798h 3019
29.237Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_36)—Offset 79Ch 3020
29.238Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_37)—Offset 7A0h 3020
29.239Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_37)—Offset 7A4h 3022
29.240Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_37)—Offset 7A8h 3024
29.241Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_37)—Offset 7ACh 3025
29.242Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_38)—Offset 7B0h 3025
29.243Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_38)—Offset 7B4h 3027
29.244Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_38)—Offset 7B8h 3029
29.245Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_38)—Offset 7BCh 3030
29.246Revision ID (REV_ID)—Offset 0h.....3030
29.247Capability List Register (CAP_LIST_0)—Offset 4h 3031
29.248Family Base Address (FAMBAR)—Offset 8h.....3031
29.249Pad Base Address (PADBAR)—Offset Ch.....3031
29.250Miscellaneous Configuration (MISCCFG)—Offset 10h.....3032
29.251Miscellaneous Secured Configuration (MISCSECCFG)—Offset 14h 3033
29.252Reserved (RSVD0[0])—Offset 18h.....3033
29.253Reserved (RSVD0[1])—Offset 1Ch 3034
29.254Pad Ownership (PAD_OWN_north_0)—Offset 20h.....3034
29.255Pad Ownership (PAD_OWN_north_1)—Offset 24h.....3038
29.256Pad Ownership (PAD_OWN_north_2)—Offset 28h.....3043
29.257Pad Ownership (PAD_OWN_north_3)—Offset 2Ch 3047
29.258Pad Ownership (PAD_OWN_north_4)—Offset 30h.....3052
29.259Pad Ownership (PAD_OWN_north_5)—Offset 34h.....3056
29.260Pad Ownership (PAD_OWN_north_6)—Offset 38h.....3061
29.261Pad Ownership (PAD_OWN_north_7)—Offset 3Ch 3065
29.262Pad Ownership (PAD_OWN_north_8)—Offset 40h.....3070
29.263Pad Ownership (PAD_OWN_north_9)—Offset 44h.....3074
29.264Reserved (RSVD1[0])—Offset 48h.....3079

29.265Reserved (RSVD1[1])—Offset 4Ch 3079
29.266Reserved (RSVD1[2])—Offset 50h 3079
29.267Reserved (RSVD1[3])—Offset 54h 3079
29.268Reserved (RSVD1[4])—Offset 58h 3080
29.269Reserved (RSVD1[5])—Offset 5Ch 3080
29.270GPI Virtual Wire Message Enable (GPI_VWE_north_0)—Offset 60h 3080
29.271GPI Virtual Wire Message Enable (GPI_VWE_north_1)—Offset 64h 3089
29.272GPI Virtual Wire Message Enable (GPI_VWE_north_2)—Offset 68h 3097
29.273Reserved (RSVD2[0])—Offset 6Ch 3101
29.274Reserved (RSVD2[1])—Offset 70h 3102
29.275Reserved (RSVD2[2])—Offset 74h 3102
29.276Reserved (RSVD2[3])—Offset 78h 3102
29.277Reserved (RSVD2[4])—Offset 7Ch 3102
29.278Pad Configuration Lock (PADCFGLOCK_north_0)—Offset 80h 3103
29.279Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_north_0)—Offset 84h .. 3119
29.280Pad Configuration Lock (PADCFGLOCK_north_1)—Offset 88h 3135
29.281Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_north_1)—Offset 8Ch .. 3151
29.282Pad Configuration Lock (PADCFGLOCK_north_2)—Offset 90h 3167
29.283Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_north_2)—Offset 94h .. 3175
29.284Reserved (RSVD3[0])—Offset 98h 3183
29.285Reserved (RSVD3[1])—Offset 9Ch 3184
29.286Reserved (RSVD3[2])—Offset A0h 3184
29.287Reserved (RSVD3[3])—Offset A4h 3184
29.288Reserved (RSVD3[4])—Offset A8h 3184
29.289Reserved (RSVD3[5])—Offset ACh 3185
29.290Host Software Pad Ownership (HOSTSW_OWN_north_0)—Offset B0h 3185
29.291Host Software Pad Ownership (HOSTSW_OWN_north_1)—Offset B4h 3191
29.292Host Software Pad Ownership (HOSTSW_OWN_north_2)—Offset B8h 3198
29.293Reserved (RSVD4[0])—Offset BCh 3201
29.294Reserved (RSVD4[1])—Offset C0h 3202
29.295Reserved (RSVD4[2])—Offset C4h 3202
29.296Reserved (RSVD4[3])—Offset C8h 3202
29.297Reserved (RSVD4[4])—Offset CCh 3203
29.298Reserved (RSVD4[5])—Offset D0h 3203
29.299Reserved (RSVD4[6])—Offset D4h 3203
29.300Reserved (RSVD4[7])—Offset D8h 3203
29.301Reserved (RSVD4[8])—Offset DCh 3204
29.302Reserved (RSVD4[9])—Offset E0h 3204
29.303Reserved (RSVD4[10])—Offset E4h 3204
29.304Reserved (RSVD4[11])—Offset E8h 3204
29.305Reserved (RSVD4[12])—Offset ECh 3205
29.306Reserved (RSVD4[13])—Offset F0h 3205
29.307Reserved (RSVD4[14])—Offset F4h 3205
29.308Reserved (RSVD4[15])—Offset F8h 3206
29.309Reserved (RSVD4[16])—Offset FCh 3206
29.310GPI Interrupt Status (GPI_IS_north_0)—Offset 100h 3206
29.311GPI Interrupt Status (GPI_IS_north_1)—Offset 104h 3214
29.312GPI Interrupt Status (GPI_IS_north_2)—Offset 108h 3222
29.313Reserved (RSVD5)—Offset 10Ch 3227
29.314GPI Interrupt Enable (GPI_IE_north_0)—Offset 110h 3227
29.315GPI Interrupt Enable (GPI_IE_north_1)—Offset 114h 3233
29.316GPI Interrupt Enable (GPI_IE_north_2)—Offset 118h 3238

29.317Reserved (RSVD6[0])—Offset 11Ch....3241
29.318Reserved (RSVD6[1])—Offset 120h....3242
29.319Reserved (RSVD6[2])—Offset 124h....3242
29.320Reserved (RSVD6[3])—Offset 128h....3242
29.321Reserved (RSVD6[4])—Offset 12Ch....3242
29.322GPI General Purpose Events Status (GPI_GPE_STS_north_0)—Offset 130h....3243
29.323GPI General Purpose Events Status (GPI_GPE_STS_north_1)—Offset 134h....3251
29.324GPI General Purpose Events Status (GPI_GPE_STS_north_2)—Offset 138h....3259
29.325Reserved (RSVD7[0])—Offset 13Ch....3264
29.326Reserved (RSVD7[1])—Offset 140h....3264
29.327Reserved (RSVD7[2])—Offset 144h....3264
29.328Reserved (RSVD7[3])—Offset 148h....3264
29.329Reserved (RSVD7[4])—Offset 14Ch....3265
29.330GPI General Purpose Events Enable (GPI_GPE_EN_north_0)—Offset 150h....3265
29.331GPI General Purpose Events Enable (GPI_GPE_EN_north_1)—Offset 154h....3272
29.332GPI General Purpose Events Enable (GPI_GPE_EN_north_2)—Offset 158h....3278
29.333Reserved (RSVD8[0])—Offset 15Ch....3282
29.334Reserved (RSVD8[1])—Offset 160h....3282
29.335Reserved (RSVD8[2])—Offset 164h....3282
29.336Reserved (RSVD8[3])—Offset 168h....3283
29.337Reserved (RSVD8[4])—Offset 16Ch....3283
29.338SMI Status (GPI_SMI_STS_north_0)—Offset 170h....3283
29.339SMI Status (GPI_SMI_STS_north_1)—Offset 174h....3294
29.340SMI Status (GPI_SMI_STS_north_2)—Offset 178h....3305
29.341Reserved (RSVD9[0])—Offset 17Ch....3311
29.342Reserved (RSVD9[1])—Offset 180h....3311
29.343Reserved (RSVD9[2])—Offset 184h....3311
29.344Reserved (RSVD9[3])—Offset 188h....3312
29.345Reserved (RSVD9[4])—Offset 18Ch....3312
29.346SMI Enable (GPI_SMI_EN_north_0)—Offset 190h....3312
29.347SMI Enable (GPI_SMI_EN_north_1)—Offset 194h....3320
29.348SMI Enable (GPI_SMI_EN_north_2)—Offset 198h....3328
29.349Reserved (RSVD10[0])—Offset 19Ch....3333
29.350Reserved (RSVD10[1])—Offset 1A0h....3333
29.351Reserved (RSVD10[2])—Offset 1A4h....3333
29.352Reserved (RSVD10[3])—Offset 1A8h....3334
29.353Reserved (RSVD10[4])—Offset 1ACh....3334
29.354NMI Status (GPI_NMI_STS_north_0)—Offset 1B0h....3334
29.355NMI Status (GPI_NMI_STS_north_1)—Offset 1B4h....3343
29.356NMI Status (GPI_NMI_STS_north_2)—Offset 1B8h....3351
29.357Reserved (RSVD11[0])—Offset 1BCh....3355
29.358Reserved (RSVD11[1])—Offset 1C0h....3355
29.359Reserved (RSVD11[2])—Offset 1C4h....3356
29.360Reserved (RSVD11[3])—Offset 1C8h....3356
29.361Reserved (RSVD11[4])—Offset 1CCh....3356
29.362NMI Enable (GPI_NMI_EN_north_0)—Offset 1D0h....3356
29.363NMI Enable (GPI_NMI_EN_north_1)—Offset 1D4h....3364
29.364NMI Enable (GPI_NMI_EN_north_2)—Offset 1D8h....3372
29.365Event Trigger Output Enable (EVOUTEN_0)—Offset 210h....3376
29.366Event Trigger Output Enable (EVOUTEN_1)—Offset 214h....3377
29.367Event Trigger Mapping (EVMAP_0)—Offset 220h....3377
29.368Event Trigger Mapping (EVMAP_1)—Offset 228h....3380
29.369Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_SVID)—Offset 300h....3384
29.370Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_SVID)—Offset 304h....3384
29.371Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_SVID)—Offset 308h....3385

29.372Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_SVID)—Offset 30Ch 3385
29.373Family Configuration Register (FAM_CFG_SVID)—Offset 310h 3386
29.374Family Reserved Register DW5 (FAM_RSVD_DW5SVID)—Offset 314h..... 3387
29.375Family Reserved Register DW6 (FAM_RSVD_DW6SVID)—Offset 318h..... 3388
29.376Family Reserved Register DW7 (FAM_RSVD_DW7SVID)—Offset 31Ch 3388
29.377Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_SPI_1P8)—Offset 320h 3388
29.378Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_SPI_1P8)—Offset 324h 3389
29.379Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_SPI_1P8)—Offset 328h 3389
29.380Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_SPI_1P8)—Offset 32Ch 3390
29.381Family Configuration Register (FAM_CFG_SPI_1P8)—Offset 330h 3390
29.382Family Reserved Register DW5 (FAM_RSVD_DW5SPI_1P8)—Offset 334h 3392
29.383Family Reserved Register DW6 (FAM_RSVD_DW6SPI_1P8)—Offset 338h 3392
29.384Family Reserved Register DW7 (FAM_RSVD_DW7SPI_1P8)—Offset 33Ch 3392
29.385Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_PMU_3P3_A)—Offset 340h . 3393
29.386Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_PMU_3P3_A)—Offset 344h . 3393
29.387Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_PMU_3P3_A)—Offset 348h . 3394
29.388Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_PMU_3P3_A)—Offset 34Ch . 3394
29.389Family Configuration Register (FAM_CFG_PMU_3P3_A)—Offset 350h..... 3394
29.390Family Reserved Register DW5 (FAM_RSVD_DW5PMU_3P3_A)—Offset 354h 3396
29.391Family Reserved Register DW6 (FAM_RSVD_DW6PMU_3P3_A)—Offset 358h 3396
29.392Family Reserved Register DW7 (FAM_RSVD_DW7PMU_3P3_A)—Offset 35Ch 3397
29.393Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_PMU_3P3_B)—Offset 360h . 3397
29.394Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_PMU_3P3_B)—Offset 364h . 3397
29.395Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_PMU_3P3_B)—Offset 368h . 3398
29.396Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_PMU_3P3_B)—Offset 36Ch . 3398
29.397Family Configuration Register (FAM_CFG_PMU_3P3_B)—Offset 370h..... 3399
29.398Family Reserved Register DW5 (FAM_RSVD_DW5PMU_3P3_B)—Offset 374h 3400
29.399Family Reserved Register DW6 (FAM_RSVD_DW6PMU_3P3_B)—Offset 378h 3401
29.400Family Reserved Register DW7 (FAM_RSVD_DW7PMU_3P3_B)—Offset 37Ch 3401
29.401Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_I2C_3P3_A)—Offset 380h... 3401
29.402Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_I2C_3P3_A)—Offset 384h... 3402
29.403Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_I2C_3P3_A)—Offset 388h... 3402
29.404Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_I2C_3P3_A)—Offset 38Ch .. 3403
29.405Family Configuration Register (FAM_CFG_I2C_3P3_A)—Offset 390h..... 3403
29.406Family Reserved Register DW5 (FAM_RSVD_DW5I2C_3P3_A)—Offset 394h 3405
29.407Family Reserved Register DW6 (FAM_RSVD_DW6I2C_3P3_A)—Offset 398h 3405
29.408Family Reserved Register DW7 (FAM_RSVD_DW7I2C_3P3_A)—Offset 39Ch 3405
29.409Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_I2C_3P3_B)—Offset 3A0h .. 3406
29.410Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_I2C_3P3_B)—Offset 3A4h .. 3406
29.411Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_I2C_3P3_B)—Offset 3A8h .. 3407
29.412Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_I2C_3P3_B)—Offset 3ACh .. 3407
29.413Family Configuration Register (FAM_CFG_I2C_3P3_B)—Offset 3B0h..... 3407
29.414Family Reserved Register DW5 (FAM_RSVD_DW5I2C_3P3_B)—Offset 3B4h 3409
29.415Family Reserved Register DW6 (FAM_RSVD_DW6I2C_3P3_B)—Offset 3B8h 3409
29.416Family Reserved Register DW7 (FAM_RSVD_DW7I2C_3P3_B)—Offset 3BCh 3410
29.417Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_PCIE_3P3)—Offset 3C0h 3410
29.418Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_PCIE_3P3)—Offset 3C4h 3410
29.419Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_PCIE_3P3)—Offset 3C8h 3411
29.420Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_PCIE_3P3)—Offset 3CCh 3411
29.421Family Configuration Register (FAM_CFG_PCIE_3P3)—Offset 3D0h 3412
29.422Family Reserved Register DW5 (FAM_RSVD_DW5PCIE_3P3)—Offset 3D4h..... 3413
29.423Family Reserved Register DW6 (FAM_RSVD_DW6PCIE_3P3)—Offset 3D8h..... 3414
29.424Family Reserved Register DW7 (FAM_RSVD_DW7PCIE_3P3)—Offset 3DCh..... 3414
29.425Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_DISP_3P3_A)—Offset 3E0h. 3414
29.426Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_DISP_3P3_A)—Offset 3E4h. 3415

29.427Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_DISP_3P3_A)—Offset 3E8h .3415
29.428Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_DISP_3P3_A)—Offset 3ECh .3416
29.429Family Configuration Register (FAM_CFG_DISP_3P3_A)—Offset 3F0h 3416
29.430Family Reserved Register DW5 (FAM_RSVD_DW5DISP_3P3_A)—Offset 3F4h 3418
29.431Family Reserved Register DW6 (FAM_RSVD_DW6DISP_3P3_A)—Offset 3F8h 3418
29.432Family Reserved Register DW7 (FAM_RSVD_DW7DISP_3P3_A)—Offset 3FCh.....3418
29.433Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_DISP_3P3_B)—Offset 400h .3419
29.434Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_DISP_3P3_B)—Offset 404h .3419
29.435Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_DISP_3P3_B)—Offset 408h .3420
29.436Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_DISP_3P3_B)—Offset 40Ch .3420
29.437Family Configuration Register (FAM_CFG_DISP_3P3_B)—Offset 410h 3420
29.438Family Reserved Register DW5 (FAM_RSVD_DW5DISP_3P3_B)—Offset 414h.....3422
29.439Family Reserved Register DW6 (FAM_RSVD_DW6DISP_3P3_B)—Offset 418h.....3422
29.440Family Reserved Register DW7 (FAM_RSVD_DW7DISP_3P3_B)—Offset 41Ch.....3423
29.441Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_GPHS3_3P3_A)—Offset 420h..... 3423
29.442Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_GPHS3_3P3_A)—Offset 424h..... 3423
29.443Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_GPHS3_3P3_A)—Offset 428h..... 3424
29.444Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_GPHS3_3P3_A)—Offset 42Ch 3424
29.445Family Configuration Register (FAM_CFG_GPHS3_3P3_A)—Offset 430h.....3425
29.446Family Reserved Register DW5 (FAM_RSVD_DW5GPHS3_3P3_A)—Offset 434h 3426
29.447Family Reserved Register DW6 (FAM_RSVD_DW6GPHS3_3P3_A)—Offset 438h 3427
29.448Family Reserved Register DW7 (FAM_RSVD_DW7GPHS3_3P3_A)—Offset 43Ch 3427
29.449Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_GPHS3_3P3_B)—Offset 440h..... 3427
29.450Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_GPHS3_3P3_B)—Offset 444h..... 3428
29.451Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_GPHS3_3P3_B)—Offset 448h 3428
29.452Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_GPHS3_3P3_B)—Offset 44Ch 3429
29.453Family Configuration Register (FAM_CFG_GPHS3_3P3_B)—Offset 450h.....3429
29.454Family Reserved Register DW5 (FAM_RSVD_DW5GPHS3_3P3_B)—Offset 454h 3431
29.455Family Reserved Register DW6 (FAM_RSVD_DW6GPHS3_3P3_B)—Offset 458h 3431
29.456Family Reserved Register DW7 (FAM_RSVD_DW7GPHS3_3P3_B)—Offset 45Ch 3431
29.457Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_LPC_3P3)—Offset 460h.....3432
29.458Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_LPC_3P3)—Offset 464h.....3432
29.459Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_LPC_3P3)—Offset 468h.....3433
29.460Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_LPC_3P3)—Offset 46Ch 3433
29.461Family Configuration Register (FAM_CFG_LPC_3P3)—Offset 470h.....3433
29.462Family Reserved Register DW5 (FAM_RSVD_DW5LPC_3P3)—Offset 474h 3435
29.463Family Reserved Register DW6 (FAM_RSVD_DW6LPC_3P3)—Offset 478h 3435
29.464Family Reserved Register DW7 (FAM_RSVD_DW7LPC_3P3)—Offset 47Ch 3436
29.465Pad Configuration DW0 (PAD_CFG_DW0_GPIO_76)—Offset 600h.....3436
29.466Pad Configuration DW1 (PAD_CFG_DW1_GPIO_76)—Offset 604h.....3438
29.467Pad Configuration DW2 (PAD_CFG_DW2_GPIO_76)—Offset 608h.....3440
29.468Pad Configuration DW3 (PAD_CFG_DW3_GPIO_76)—Offset 60Ch.....3441
29.469Pad Configuration DW0 (PAD_CFG_DW0_GPIO_77)—Offset 610h.....3441
29.470Pad Configuration DW1 (PAD_CFG_DW1_GPIO_77)—Offset 614h.....3443
29.471Pad Configuration DW2 (PAD_CFG_DW2_GPIO_77)—Offset 618h.....3445
29.472Pad Configuration DW3 (PAD_CFG_DW3_GPIO_77)—Offset 61Ch.....3446
29.473Pad Configuration DW0 (PAD_CFG_DW0_GPIO_78)—Offset 620h.....3446

29.474Pad Configuration DW1 (PAD_CFG_DW1_GPIO_78)—Offset 624h 3448
29.475Pad Configuration DW2 (PAD_CFG_DW2_GPIO_78)—Offset 628h 3450
29.476Pad Configuration DW3 (PAD_CFG_DW3_GPIO_78)—Offset 62Ch 3451
29.477Pad Configuration DW0 (PAD_CFG_DW0_GPIO_79)—Offset 630h 3451
29.478Pad Configuration DW1 (PAD_CFG_DW1_GPIO_79)—Offset 634h 3453
29.479Pad Configuration DW2 (PAD_CFG_DW2_GPIO_79)—Offset 638h 3455
29.480Pad Configuration DW3 (PAD_CFG_DW3_GPIO_79)—Offset 63Ch 3456
29.481Pad Configuration DW0 (PAD_CFG_DW0_GPIO_80)—Offset 640h 3456
29.482Pad Configuration DW1 (PAD_CFG_DW1_GPIO_80)—Offset 644h 3458
29.483Pad Configuration DW2 (PAD_CFG_DW2_GPIO_80)—Offset 648h 3460
29.484Pad Configuration DW3 (PAD_CFG_DW3_GPIO_80)—Offset 64Ch 3461
29.485Pad Configuration DW0 (PAD_CFG_DW0_GPIO_81)—Offset 650h 3461
29.486Pad Configuration DW1 (PAD_CFG_DW1_GPIO_81)—Offset 654h 3463
29.487Pad Configuration DW2 (PAD_CFG_DW2_GPIO_81)—Offset 658h 3465
29.488Pad Configuration DW3 (PAD_CFG_DW3_GPIO_81)—Offset 65Ch 3466
29.489Pad Configuration DW0 (PAD_CFG_DW0_GPIO_82)—Offset 660h 3466
29.490Pad Configuration DW1 (PAD_CFG_DW1_GPIO_82)—Offset 664h 3468
29.491Pad Configuration DW2 (PAD_CFG_DW2_GPIO_82)—Offset 668h 3470
29.492Pad Configuration DW3 (PAD_CFG_DW3_GPIO_82)—Offset 66Ch 3471
29.493Pad Configuration DW0 (PAD_CFG_DW0_GPIO_83)—Offset 670h 3471
29.494Pad Configuration DW1 (PAD_CFG_DW1_GPIO_83)—Offset 674h 3473
29.495Pad Configuration DW2 (PAD_CFG_DW2_GPIO_83)—Offset 678h 3475
29.496Pad Configuration DW3 (PAD_CFG_DW3_GPIO_83)—Offset 67Ch 3476
29.497Pad Configuration DW0 (PAD_CFG_DW0_GPIO_84)—Offset 680h 3476
29.498Pad Configuration DW1 (PAD_CFG_DW1_GPIO_84)—Offset 684h 3478
29.499Pad Configuration DW2 (PAD_CFG_DW2_GPIO_84)—Offset 688h 3480
29.500Pad Configuration DW3 (PAD_CFG_DW3_GPIO_84)—Offset 68Ch 3481
29.501Pad Configuration DW0 (PAD_CFG_DW0_GPIO_85)—Offset 690h 3481
29.502Pad Configuration DW1 (PAD_CFG_DW1_GPIO_85)—Offset 694h 3483
29.503Pad Configuration DW2 (PAD_CFG_DW2_GPIO_85)—Offset 698h 3485
29.504Pad Configuration DW3 (PAD_CFG_DW3_GPIO_85)—Offset 69Ch 3486
29.505Pad Configuration DW0 (PAD_CFG_DW0_GPIO_86)—Offset 6A0h 3486
29.506Pad Configuration DW1 (PAD_CFG_DW1_GPIO_86)—Offset 6A4h 3488
29.507Pad Configuration DW2 (PAD_CFG_DW2_GPIO_86)—Offset 6A8h 3490
29.508Pad Configuration DW3 (PAD_CFG_DW3_GPIO_86)—Offset 6ACh 3491
29.509Pad Configuration DW0 (PAD_CFG_DW0_GPIO_87)—Offset 6B0h 3491
29.510Pad Configuration DW1 (PAD_CFG_DW1_GPIO_87)—Offset 6B4h 3493
29.511Pad Configuration DW2 (PAD_CFG_DW2_GPIO_87)—Offset 6B8h 3495
29.512Pad Configuration DW3 (PAD_CFG_DW3_GPIO_87)—Offset 6BCh 3496
29.513Pad Configuration DW0 (PAD_CFG_DW0_GPIO_88)—Offset 6C0h 3496
29.514Pad Configuration DW1 (PAD_CFG_DW1_GPIO_88)—Offset 6C4h 3498
29.515Pad Configuration DW2 (PAD_CFG_DW2_GPIO_88)—Offset 6C8h 3500
29.516Pad Configuration DW3 (PAD_CFG_DW3_GPIO_88)—Offset 6CCh 3501
29.517Pad Configuration DW0 (PAD_CFG_DW0_GPIO_89)—Offset 6D0h 3501
29.518Pad Configuration DW1 (PAD_CFG_DW1_GPIO_89)—Offset 6D4h 3503
29.519Pad Configuration DW2 (PAD_CFG_DW2_GPIO_89)—Offset 6D8h 3505
29.520Pad Configuration DW3 (PAD_CFG_DW3_GPIO_89)—Offset 6DCh 3506
29.521Pad Configuration DW0 (PAD_CFG_DW0_GPIO_90)—Offset 6E0h 3506
29.522Pad Configuration DW1 (PAD_CFG_DW1_GPIO_90)—Offset 6E4h 3508
29.523Pad Configuration DW2 (PAD_CFG_DW2_GPIO_90)—Offset 6E8h 3510
29.524Pad Configuration DW3 (PAD_CFG_DW3_GPIO_90)—Offset 6ECh 3511
29.525Pad Configuration DW0 (PAD_CFG_DW0_GPIO_91)—Offset 6F0h 3511
29.526Pad Configuration DW1 (PAD_CFG_DW1_GPIO_91)—Offset 6F4h 3513
29.527Pad Configuration DW2 (PAD_CFG_DW2_GPIO_91)—Offset 6F8h 3515
29.528Pad Configuration DW3 (PAD_CFG_DW3_GPIO_91)—Offset 6FCh 3516

29.529Pad Configuration DW0 (PAD_CFG_DW0_GPIO_92)—Offset 700h 3516
29.530Pad Configuration DW1 (PAD_CFG_DW1_GPIO_92)—Offset 704h 3518
29.531Pad Configuration DW2 (PAD_CFG_DW2_GPIO_92)—Offset 708h 3520
29.532Pad Configuration DW3 (PAD_CFG_DW3_GPIO_92)—Offset 70Ch 3521
29.533Pad Configuration DW0 (PAD_CFG_DW0_GPIO_93)—Offset 710h 3521
29.534Pad Configuration DW1 (PAD_CFG_DW1_GPIO_93)—Offset 714h 3523
29.535Pad Configuration DW2 (PAD_CFG_DW2_GPIO_93)—Offset 718h 3525
29.536Pad Configuration DW3 (PAD_CFG_DW3_GPIO_93)—Offset 71Ch 3526
29.537Pad Configuration DW0 (PAD_CFG_DW0_GPIO_94)—Offset 720h 3526
29.538Pad Configuration DW1 (PAD_CFG_DW1_GPIO_94)—Offset 724h 3528
29.539Pad Configuration DW2 (PAD_CFG_DW2_GPIO_94)—Offset 728h 3530
29.540Pad Configuration DW3 (PAD_CFG_DW3_GPIO_94)—Offset 72Ch 3531
29.541Pad Configuration DW0 (PAD_CFG_DW0_GPIO_95)—Offset 730h 3531
29.542Pad Configuration DW1 (PAD_CFG_DW1_GPIO_95)—Offset 734h 3533
29.543Pad Configuration DW2 (PAD_CFG_DW2_GPIO_95)—Offset 738h 3535
29.544Pad Configuration DW3 (PAD_CFG_DW3_GPIO_95)—Offset 73Ch 3536
29.545Pad Configuration DW0 (PAD_CFG_DW0_GPIO_96)—Offset 740h 3536
29.546Pad Configuration DW1 (PAD_CFG_DW1_GPIO_96)—Offset 744h 3538
29.547Pad Configuration DW2 (PAD_CFG_DW2_GPIO_96)—Offset 748h 3540
29.548Pad Configuration DW3 (PAD_CFG_DW3_GPIO_96)—Offset 74Ch 3541
29.549Pad Configuration DW0 (PAD_CFG_DW0_GPIO_97)—Offset 750h 3541
29.550Pad Configuration DW1 (PAD_CFG_DW1_GPIO_97)—Offset 754h 3543
29.551Pad Configuration DW2 (PAD_CFG_DW2_GPIO_97)—Offset 758h 3545
29.552Pad Configuration DW3 (PAD_CFG_DW3_GPIO_97)—Offset 75Ch 3546
29.553Pad Configuration DW0 (PAD_CFG_DW0_GPIO_98)—Offset 760h 3546
29.554Pad Configuration DW1 (PAD_CFG_DW1_GPIO_98)—Offset 764h 3548
29.555Pad Configuration DW2 (PAD_CFG_DW2_GPIO_98)—Offset 768h 3550
29.556Pad Configuration DW3 (PAD_CFG_DW3_GPIO_98)—Offset 76Ch 3551
29.557Pad Configuration DW0 (PAD_CFG_DW0_GPIO_99)—Offset 770h 3551
29.558Pad Configuration DW1 (PAD_CFG_DW1_GPIO_99)—Offset 774h 3553
29.559Pad Configuration DW2 (PAD_CFG_DW2_GPIO_99)—Offset 778h 3555
29.560Pad Configuration DW3 (PAD_CFG_DW3_GPIO_99)—Offset 77Ch 3556
29.561Pad Configuration DW0 (PAD_CFG_DW0_GPIO_100)—Offset 780h 3556
29.562Pad Configuration DW1 (PAD_CFG_DW1_GPIO_100)—Offset 784h 3558
29.563Pad Configuration DW2 (PAD_CFG_DW2_GPIO_100)—Offset 788h 3560
29.564Pad Configuration DW3 (PAD_CFG_DW3_GPIO_100)—Offset 78Ch 3561
29.565Pad Configuration DW0 (PAD_CFG_DW0_GPIO_101)—Offset 790h 3561
29.566Pad Configuration DW1 (PAD_CFG_DW1_GPIO_101)—Offset 794h 3563
29.567Pad Configuration DW2 (PAD_CFG_DW2_GPIO_101)—Offset 798h 3565
29.568Pad Configuration DW3 (PAD_CFG_DW3_GPIO_101)—Offset 79Ch 3566
29.569Pad Configuration DW0 (PAD_CFG_DW0_GPIO_102)—Offset 7A0h 3566
29.570Pad Configuration DW1 (PAD_CFG_DW1_GPIO_102)—Offset 7A4h 3568
29.571Pad Configuration DW2 (PAD_CFG_DW2_GPIO_102)—Offset 7A8h 3570
29.572Pad Configuration DW3 (PAD_CFG_DW3_GPIO_102)—Offset 7ACh 3571
29.573Pad Configuration DW0 (PAD_CFG_DW0_GPIO_103)—Offset 7B0h 3571
29.574Pad Configuration DW1 (PAD_CFG_DW1_GPIO_103)—Offset 7B4h 3573
29.575Pad Configuration DW2 (PAD_CFG_DW2_GPIO_103)—Offset 7B8h 3575
29.576Pad Configuration DW3 (PAD_CFG_DW3_GPIO_103)—Offset 7BCh 3576
29.577Pad Configuration DW0 (PAD_CFG_DW0_GPIO_104)—Offset 7C0h 3576
29.578Pad Configuration DW1 (PAD_CFG_DW1_GPIO_104)—Offset 7C4h 3578
29.579Pad Configuration DW2 (PAD_CFG_DW2_GPIO_104)—Offset 7C8h 3580
29.580Pad Configuration DW3 (PAD_CFG_DW3_GPIO_104)—Offset 7CCh 3581
29.581Pad Configuration DW0 (PAD_CFG_DW0_GPIO_105)—Offset 7D0h 3581
29.582Pad Configuration DW1 (PAD_CFG_DW1_GPIO_105)—Offset 7D4h 3583
29.583Pad Configuration DW2 (PAD_CFG_DW2_GPIO_105)—Offset 7D8h 3585

29.584Pad Configuration DW3 (PAD_CFG_DW3_GPIO_105)—Offset 7DCh 3586
29.585Pad Configuration DW0 (PAD_CFG_DW0_GPIO_106)—Offset 7E0h..... 3586
29.586Pad Configuration DW1 (PAD_CFG_DW1_GPIO_106)—Offset 7E4h..... 3588
29.587Pad Configuration DW2 (PAD_CFG_DW2_GPIO_106)—Offset 7E8h..... 3590
29.588Pad Configuration DW3 (PAD_CFG_DW3_GPIO_106)—Offset 7ECh 3591
29.589Pad Configuration DW0 (PAD_CFG_DW0_GPIO_107)—Offset 7F0h..... 3591
29.590Pad Configuration DW1 (PAD_CFG_DW1_GPIO_107)—Offset 7F4h..... 3593
29.591Pad Configuration DW2 (PAD_CFG_DW2_GPIO_107)—Offset 7F8h..... 3595
29.592Pad Configuration DW3 (PAD_CFG_DW3_GPIO_107)—Offset 7FCh..... 3596
29.593Pad Configuration DW0 (PAD_CFG_DW0_GPIO_108)—Offset 800h..... 3596
29.594Pad Configuration DW1 (PAD_CFG_DW1_GPIO_108)—Offset 804h..... 3598
29.595Pad Configuration DW2 (PAD_CFG_DW2_GPIO_108)—Offset 808h..... 3600
29.596Pad Configuration DW3 (PAD_CFG_DW3_GPIO_108)—Offset 80Ch 3601
29.597Pad Configuration DW0 (PAD_CFG_DW0_GPIO_109)—Offset 810h..... 3601
29.598Pad Configuration DW1 (PAD_CFG_DW1_GPIO_109)—Offset 814h..... 3603
29.599Pad Configuration DW2 (PAD_CFG_DW2_GPIO_109)—Offset 818h..... 3605
29.600Pad Configuration DW3 (PAD_CFG_DW3_GPIO_109)—Offset 81Ch 3606
29.601Pad Configuration DW0 (PAD_CFG_DW0_GPIO_110)—Offset 820h..... 3606
29.602Pad Configuration DW1 (PAD_CFG_DW1_GPIO_110)—Offset 824h..... 3608
29.603Pad Configuration DW2 (PAD_CFG_DW2_GPIO_110)—Offset 828h..... 3610
29.604Pad Configuration DW3 (PAD_CFG_DW3_GPIO_110)—Offset 82Ch 3611
29.605Pad Configuration DW0 (PAD_CFG_DW0_GPIO_111)—Offset 830h..... 3611
29.606Pad Configuration DW1 (PAD_CFG_DW1_GPIO_111)—Offset 834h..... 3613
29.607Pad Configuration DW2 (PAD_CFG_DW2_GPIO_111)—Offset 838h..... 3615
29.608Pad Configuration DW3 (PAD_CFG_DW3_GPIO_111)—Offset 83Ch 3616
29.609Pad Configuration DW0 (PAD_CFG_DW0_GPIO_112)—Offset 840h..... 3616
29.610Pad Configuration DW1 (PAD_CFG_DW1_GPIO_112)—Offset 844h..... 3618
29.611Pad Configuration DW2 (PAD_CFG_DW2_GPIO_112)—Offset 848h..... 3620
29.612Pad Configuration DW3 (PAD_CFG_DW3_GPIO_112)—Offset 84Ch 3621
29.613Pad Configuration DW0 (PAD_CFG_DW0_GPIO_113)—Offset 850h..... 3621
29.614Pad Configuration DW1 (PAD_CFG_DW1_GPIO_113)—Offset 854h..... 3623
29.615Pad Configuration DW2 (PAD_CFG_DW2_GPIO_113)—Offset 858h..... 3625
29.616Pad Configuration DW3 (PAD_CFG_DW3_GPIO_113)—Offset 85Ch 3626
29.617Pad Configuration DW0 (PAD_CFG_DW0_GPIO_114)—Offset 860h..... 3626
29.618Pad Configuration DW1 (PAD_CFG_DW1_GPIO_114)—Offset 864h..... 3628
29.619Pad Configuration DW2 (PAD_CFG_DW2_GPIO_114)—Offset 868h..... 3630
29.620Pad Configuration DW3 (PAD_CFG_DW3_GPIO_114)—Offset 86Ch 3631
29.621Pad Configuration DW0 (PAD_CFG_DW0_GPIO_115)—Offset 870h..... 3631
29.622Pad Configuration DW1 (PAD_CFG_DW1_GPIO_115)—Offset 874h..... 3633
29.623Pad Configuration DW2 (PAD_CFG_DW2_GPIO_115)—Offset 878h..... 3635
29.624Pad Configuration DW3 (PAD_CFG_DW3_GPIO_115)—Offset 87Ch 3636
29.625Pad Configuration DW0 (PAD_CFG_DW0_GPIO_116)—Offset 880h..... 3636
29.626Pad Configuration DW1 (PAD_CFG_DW1_GPIO_116)—Offset 884h..... 3638
29.627Pad Configuration DW2 (PAD_CFG_DW2_GPIO_116)—Offset 888h..... 3640
29.628Pad Configuration DW3 (PAD_CFG_DW3_GPIO_116)—Offset 88Ch 3641
29.629Pad Configuration DW0 (PAD_CFG_DW0_GPIO_117)—Offset 890h..... 3641
29.630Pad Configuration DW1 (PAD_CFG_DW1_GPIO_117)—Offset 894h..... 3643
29.631Pad Configuration DW2 (PAD_CFG_DW2_GPIO_117)—Offset 898h..... 3645
29.632Pad Configuration DW3 (PAD_CFG_DW3_GPIO_117)—Offset 89Ch 3646
29.633Pad Configuration DW0 (PAD_CFG_DW0_GPIO_118)—Offset 8A0h 3646
29.634Pad Configuration DW1 (PAD_CFG_DW1_GPIO_118)—Offset 8A4h 3648
29.635Pad Configuration DW2 (PAD_CFG_DW2_GPIO_118)—Offset 8A8h 3650
29.636Pad Configuration DW3 (PAD_CFG_DW3_GPIO_118)—Offset 8ACh 3651
29.637Pad Configuration DW0 (PAD_CFG_DW0_GPIO_119)—Offset 8B0h 3651
29.638Pad Configuration DW1 (PAD_CFG_DW1_GPIO_119)—Offset 8B4h 3653

29.639Pad Configuration DW2 (PAD_CFG_DW2_GPIO_119)—Offset 8B8h 3655
29.640Pad Configuration DW3 (PAD_CFG_DW3_GPIO_119)—Offset 8BCh 3656
29.641Pad Configuration DW0 (PAD_CFG_DW0_GPIO_120)—Offset 8C0h 3656
29.642Pad Configuration DW1 (PAD_CFG_DW1_GPIO_120)—Offset 8C4h 3658
29.643Pad Configuration DW2 (PAD_CFG_DW2_GPIO_120)—Offset 8C8h 3660
29.644Pad Configuration DW3 (PAD_CFG_DW3_GPIO_120)—Offset 8CCh 3661
29.645Pad Configuration DW0 (PAD_CFG_DW0_GPIO_121)—Offset 8D0h 3661
29.646Pad Configuration DW1 (PAD_CFG_DW1_GPIO_121)—Offset 8D4h 3663
29.647Pad Configuration DW2 (PAD_CFG_DW2_GPIO_121)—Offset 8D8h 3665
29.648Pad Configuration DW3 (PAD_CFG_DW3_GPIO_121)—Offset 8DCh 3666
29.649Pad Configuration DW0 (PAD_CFG_DW0_GPIO_122)—Offset 8E0h 3666
29.650Pad Configuration DW1 (PAD_CFG_DW1_GPIO_122)—Offset 8E4h 3668
29.651Pad Configuration DW2 (PAD_CFG_DW2_GPIO_122)—Offset 8E8h 3670
29.652Pad Configuration DW3 (PAD_CFG_DW3_GPIO_122)—Offset 8ECh 3671
29.653Pad Configuration DW0 (PAD_CFG_DW0_GPIO_123)—Offset 8F0h 3671
29.654Pad Configuration DW1 (PAD_CFG_DW1_GPIO_123)—Offset 8F4h 3673
29.655Pad Configuration DW2 (PAD_CFG_DW2_GPIO_123)—Offset 8F8h 3675
29.656Pad Configuration DW3 (PAD_CFG_DW3_GPIO_123)—Offset 8FCh 3676
29.657Pad Configuration DW0 (PAD_CFG_DW0_GPIO_124)—Offset 900h 3676
29.658Pad Configuration DW1 (PAD_CFG_DW1_GPIO_124)—Offset 904h 3678
29.659Pad Configuration DW2 (PAD_CFG_DW2_GPIO_124)—Offset 908h 3680
29.660Pad Configuration DW3 (PAD_CFG_DW3_GPIO_124)—Offset 90Ch 3681
29.661Pad Configuration DW0 (PAD_CFG_DW0_GPIO_125)—Offset 910h 3681
29.662Pad Configuration DW1 (PAD_CFG_DW1_GPIO_125)—Offset 914h 3683
29.663Pad Configuration DW2 (PAD_CFG_DW2_GPIO_125)—Offset 918h 3685
29.664Pad Configuration DW3 (PAD_CFG_DW3_GPIO_125)—Offset 91Ch 3686
29.665Pad Configuration DW0 (PAD_CFG_DW0_GPIO_126)—Offset 920h 3686
29.666Pad Configuration DW1 (PAD_CFG_DW1_GPIO_126)—Offset 924h 3688
29.667Pad Configuration DW2 (PAD_CFG_DW2_GPIO_126)—Offset 928h 3690
29.668Pad Configuration DW3 (PAD_CFG_DW3_GPIO_126)—Offset 92Ch 3691
29.669Pad Configuration DW0 (PAD_CFG_DW0_GPIO_127)—Offset 930h 3691
29.670Pad Configuration DW1 (PAD_CFG_DW1_GPIO_127)—Offset 934h 3693
29.671Pad Configuration DW2 (PAD_CFG_DW2_GPIO_127)—Offset 938h 3695
29.672Pad Configuration DW3 (PAD_CFG_DW3_GPIO_127)—Offset 93Ch 3696
29.673Pad Configuration DW0 (PAD_CFG_DW0_GPIO_128)—Offset 940h 3696
29.674Pad Configuration DW1 (PAD_CFG_DW1_GPIO_128)—Offset 944h 3698
29.675Pad Configuration DW2 (PAD_CFG_DW2_GPIO_128)—Offset 948h 3700
29.676Pad Configuration DW3 (PAD_CFG_DW3_GPIO_128)—Offset 94Ch 3701
29.677Pad Configuration DW0 (PAD_CFG_DW0_GPIO_129)—Offset 950h 3701
29.678Pad Configuration DW1 (PAD_CFG_DW1_GPIO_129)—Offset 954h 3703
29.679Pad Configuration DW2 (PAD_CFG_DW2_GPIO_129)—Offset 958h 3705
29.680Pad Configuration DW3 (PAD_CFG_DW3_GPIO_129)—Offset 95Ch 3706
29.681Pad Configuration DW0 (PAD_CFG_DW0_GPIO_130)—Offset 960h 3706
29.682Pad Configuration DW1 (PAD_CFG_DW1_GPIO_130)—Offset 964h 3708
29.683Pad Configuration DW2 (PAD_CFG_DW2_GPIO_130)—Offset 968h 3710
29.684Pad Configuration DW3 (PAD_CFG_DW3_GPIO_130)—Offset 96Ch 3711
29.685Pad Configuration DW0 (PAD_CFG_DW0_GPIO_131)—Offset 970h 3711
29.686Pad Configuration DW1 (PAD_CFG_DW1_GPIO_131)—Offset 974h 3713
29.687Pad Configuration DW2 (PAD_CFG_DW2_GPIO_131)—Offset 978h 3715
29.688Pad Configuration DW3 (PAD_CFG_DW3_GPIO_131)—Offset 97Ch 3716
29.689Pad Configuration DW0 (PAD_CFG_DW0_GPIO_132)—Offset 980h 3716
29.690Pad Configuration DW1 (PAD_CFG_DW1_GPIO_132)—Offset 984h 3718
29.691Pad Configuration DW2 (PAD_CFG_DW2_GPIO_132)—Offset 988h 3720
29.692Pad Configuration DW3 (PAD_CFG_DW3_GPIO_132)—Offset 98Ch 3721
29.693Pad Configuration DW0 (PAD_CFG_DW0_GPIO_133)—Offset 990h 3721

29.694Pad Configuration DW1 (PAD_CFG_DW1_GPIO_133)—Offset 994h..... 3723
29.695Pad Configuration DW2 (PAD_CFG_DW2_GPIO_133)—Offset 998h..... 3725
29.696Pad Configuration DW3 (PAD_CFG_DW3_GPIO_133)—Offset 99Ch 3726
29.697Pad Configuration DW0 (PAD_CFG_DW0_GPIO_134)—Offset 9A0h 3726
29.698Pad Configuration DW1 (PAD_CFG_DW1_GPIO_134)—Offset 9A4h 3728
29.699Pad Configuration DW2 (PAD_CFG_DW2_GPIO_134)—Offset 9A8h 3730
29.700Pad Configuration DW3 (PAD_CFG_DW3_GPIO_134)—Offset 9ACh 3731
29.701Pad Configuration DW0 (PAD_CFG_DW0_GPIO_135)—Offset 9B0h 3731
29.702Pad Configuration DW1 (PAD_CFG_DW1_GPIO_135)—Offset 9B4h 3733
29.703Pad Configuration DW2 (PAD_CFG_DW2_GPIO_135)—Offset 9B8h 3735
29.704Pad Configuration DW3 (PAD_CFG_DW3_GPIO_135)—Offset 9BCh 3736
29.705Pad Configuration DW0 (PAD_CFG_DW0_GPIO_136)—Offset 9C0h 3736
29.706Pad Configuration DW1 (PAD_CFG_DW1_GPIO_136)—Offset 9C4h 3738
29.707Pad Configuration DW2 (PAD_CFG_DW2_GPIO_136)—Offset 9C8h 3740
29.708Pad Configuration DW3 (PAD_CFG_DW3_GPIO_136)—Offset 9CCh 3741
29.709Pad Configuration DW0 (PAD_CFG_DW0_GPIO_137)—Offset 9D0h 3741
29.710Pad Configuration DW1 (PAD_CFG_DW1_GPIO_137)—Offset 9D4h 3743
29.711Pad Configuration DW2 (PAD_CFG_DW2_GPIO_137)—Offset 9D8h 3745
29.712Pad Configuration DW3 (PAD_CFG_DW3_GPIO_137)—Offset 9DCh 3746
29.713Pad Configuration DW0 (PAD_CFG_DW0_GPIO_138)—Offset 9E0h..... 3746
29.714Pad Configuration DW1 (PAD_CFG_DW1_GPIO_138)—Offset 9E4h..... 3748
29.715Pad Configuration DW2 (PAD_CFG_DW2_GPIO_138)—Offset 9E8h..... 3750
29.716Pad Configuration DW3 (PAD_CFG_DW3_GPIO_138)—Offset 9ECh 3751
29.717Pad Configuration DW0 (PAD_CFG_DW0_GPIO_139)—Offset 9F0h..... 3751
29.718Pad Configuration DW1 (PAD_CFG_DW1_GPIO_139)—Offset 9F4h..... 3753
29.719Pad Configuration DW2 (PAD_CFG_DW2_GPIO_139)—Offset 9F8h..... 3755
29.720Pad Configuration DW3 (PAD_CFG_DW3_GPIO_139)—Offset 9FCh..... 3756
29.721Pad Configuration DW0 (PAD_CFG_DW0_GPIO_140)—Offset A00h 3756
29.722Pad Configuration DW1 (PAD_CFG_DW1_GPIO_140)—Offset A04h 3758
29.723Pad Configuration DW2 (PAD_CFG_DW2_GPIO_140)—Offset A08h 3760
29.724Pad Configuration DW3 (PAD_CFG_DW3_GPIO_140)—Offset A0Ch 3761
29.725Pad Configuration DW0 (PAD_CFG_DW0_GPIO_141)—Offset A10h 3761
29.726Pad Configuration DW1 (PAD_CFG_DW1_GPIO_141)—Offset A14h 3763
29.727Pad Configuration DW2 (PAD_CFG_DW2_GPIO_141)—Offset A18h 3765
29.728Pad Configuration DW3 (PAD_CFG_DW3_GPIO_141)—Offset A1Ch 3766
29.729Pad Configuration DW0 (PAD_CFG_DW0_GPIO_142)—Offset A20h 3766
29.730Pad Configuration DW1 (PAD_CFG_DW1_GPIO_142)—Offset A24h 3768
29.731Pad Configuration DW2 (PAD_CFG_DW2_GPIO_142)—Offset A28h 3770
29.732Pad Configuration DW3 (PAD_CFG_DW3_GPIO_142)—Offset A2Ch 3771
29.733Pad Configuration DW0 (PAD_CFG_DW0_GPIO_143)—Offset A30h 3771
29.734Pad Configuration DW1 (PAD_CFG_DW1_GPIO_143)—Offset A34h 3773
29.735Pad Configuration DW2 (PAD_CFG_DW2_GPIO_143)—Offset A38h 3775
29.736Pad Configuration DW3 (PAD_CFG_DW3_GPIO_143)—Offset A3Ch 3776
29.737Pad Configuration DW0 (PAD_CFG_DW0_GPIO_144)—Offset A40h 3776
29.738Pad Configuration DW1 (PAD_CFG_DW1_GPIO_144)—Offset A44h 3778
29.739Pad Configuration DW2 (PAD_CFG_DW2_GPIO_144)—Offset A48h 3780
29.740Pad Configuration DW3 (PAD_CFG_DW3_GPIO_144)—Offset A4Ch 3781
29.741Pad Configuration DW0 (PAD_CFG_DW0_GPIO_145)—Offset A50h 3781
29.742Pad Configuration DW1 (PAD_CFG_DW1_GPIO_145)—Offset A54h 3783
29.743Pad Configuration DW2 (PAD_CFG_DW2_GPIO_145)—Offset A58h 3785
29.744Pad Configuration DW3 (PAD_CFG_DW3_GPIO_145)—Offset A5Ch 3786
29.745Pad Configuration DW0 (PAD_CFG_DW0_GPIO_146)—Offset A60h 3786
29.746Pad Configuration DW1 (PAD_CFG_DW1_GPIO_146)—Offset A64h 3788
29.747Pad Configuration DW2 (PAD_CFG_DW2_GPIO_146)—Offset A68h 3790
29.748Pad Configuration DW3 (PAD_CFG_DW3_GPIO_146)—Offset A6Ch 3791

29.749Pad Configuration DW0 (PAD_CFG_DW0_GPIO_147)—Offset A70h 3791
29.750Pad Configuration DW1 (PAD_CFG_DW1_GPIO_147)—Offset A74h 3793
29.751Pad Configuration DW2 (PAD_CFG_DW2_GPIO_147)—Offset A78h 3795
29.752Pad Configuration DW3 (PAD_CFG_DW3_GPIO_147)—Offset A7Ch 3796
29.753Pad Configuration DW0 (PAD_CFG_DW0_GPIO_148)—Offset A80h 3796
29.754Pad Configuration DW1 (PAD_CFG_DW1_GPIO_148)—Offset A84h 3798
29.755Pad Configuration DW2 (PAD_CFG_DW2_GPIO_148)—Offset A88h 3800
29.756Pad Configuration DW3 (PAD_CFG_DW3_GPIO_148)—Offset A8Ch 3801
29.757Pad Configuration DW0 (PAD_CFG_DW0_GPIO_149)—Offset A90h 3801
29.758Pad Configuration DW1 (PAD_CFG_DW1_GPIO_149)—Offset A94h 3803
29.759Pad Configuration DW2 (PAD_CFG_DW2_GPIO_149)—Offset A98h 3805
29.760Pad Configuration DW3 (PAD_CFG_DW3_GPIO_149)—Offset A9Ch 3806
29.761Pad Configuration DW0 (PAD_CFG_DW0_GPIO_150)—Offset AA0h 3806
29.762Pad Configuration DW1 (PAD_CFG_DW1_GPIO_150)—Offset AA4h 3808
29.763Pad Configuration DW2 (PAD_CFG_DW2_GPIO_150)—Offset AA8h 3810
29.764Pad Configuration DW3 (PAD_CFG_DW3_GPIO_150)—Offset AACh 3811
29.765Pad Configuration DW0 (PAD_CFG_DW0_GPIO_151)—Offset AB0h 3811
29.766Pad Configuration DW1 (PAD_CFG_DW1_GPIO_151)—Offset AB4h 3813
29.767Pad Configuration DW2 (PAD_CFG_DW2_GPIO_151)—Offset AB8h 3815
29.768Pad Configuration DW3 (PAD_CFG_DW3_GPIO_151)—Offset ABCh 3816
29.769Pad Configuration DW0 (PAD_CFG_DW0_GPIO_152)—Offset AC0h 3816
29.770Pad Configuration DW1 (PAD_CFG_DW1_GPIO_152)—Offset AC4h 3818
29.771Pad Configuration DW2 (PAD_CFG_DW2_GPIO_152)—Offset AC8h 3820
29.772Pad Configuration DW3 (PAD_CFG_DW3_GPIO_152)—Offset ACCh 3821
29.773Pad Configuration DW0 (PAD_CFG_DW0_GPIO_153)—Offset AD0h 3821
29.774Pad Configuration DW1 (PAD_CFG_DW1_GPIO_153)—Offset AD4h 3823
29.775Pad Configuration DW2 (PAD_CFG_DW2_GPIO_153)—Offset AD8h 3825
29.776Pad Configuration DW3 (PAD_CFG_DW3_GPIO_153)—Offset ADCh 3826
29.777Pad Configuration DW0 (PAD_CFG_DW0_GPIO_154)—Offset AE0h 3826
29.778Pad Configuration DW1 (PAD_CFG_DW1_GPIO_154)—Offset AE4h 3828
29.779Pad Configuration DW2 (PAD_CFG_DW2_GPIO_154)—Offset AE8h 3830
29.780Pad Configuration DW3 (PAD_CFG_DW3_GPIO_154)—Offset AECh 3831
29.781Pad Configuration DW0 (PAD_CFG_DW0_GPIO_155)—Offset AF0h 3831
29.782Pad Configuration DW1 (PAD_CFG_DW1_GPIO_155)—Offset AF4h 3833
29.783Pad Configuration DW2 (PAD_CFG_DW2_GPIO_155)—Offset AF8h 3835
29.784Pad Configuration DW3 (PAD_CFG_DW3_GPIO_155)—Offset AFCh 3836
29.785Revision ID (REV_ID)—Offset 0h.....3836
29.786Capability List Register (CAP_LIST_0)—Offset 4h 3837
29.787Family Base Address (FAMBAR)—Offset 8h.....3837
29.788Pad Base Address (PADBAR)—Offset Ch.....3837
29.789Miscellaneous Configuration (MISCCFG)—Offset 10h.....3838
29.790Miscellaneous Secured Configuration (MISCSECCFG)—Offset 14h 3839
29.791Reserved (RSVD0[0])—Offset 18h.....3839
29.792Reserved (RSVD0[1])—Offset 1Ch 3840
29.793Pad Ownership (PAD_OWN_northwest_0)—Offset 20h 3840
29.794Pad Ownership (PAD_OWN_northwest_1)—Offset 24h 3844
29.795Pad Ownership (PAD_OWN_northwest_2)—Offset 28h 3849
29.796Pad Ownership (PAD_OWN_northwest_3)—Offset 2Ch 3853
29.797Pad Ownership (PAD_OWN_northwest_4)—Offset 30h 3858
29.798Pad Ownership (PAD_OWN_northwest_5)—Offset 34h 3862
29.799Pad Ownership (PAD_OWN_northwest_6)—Offset 38h 3867
29.800Pad Ownership (PAD_OWN_northwest_7)—Offset 3Ch 3871
29.801Pad Ownership (PAD_OWN_northwest_8)—Offset 40h 3876
29.802Pad Ownership (PAD_OWN_northwest_9)—Offset 44h 3880
29.803Pad Ownership (PAD_OWN_northwest_10)—Offset 48h 3885

29.804Pad Ownership (PAD_OWN_northwest_11)—Offset 4Ch 3889
29.805Pad Ownership (PAD_OWN_northwest_12)—Offset 50h 3894
29.806Pad Ownership (PAD_OWN_northwest_13)—Offset 54h 3898
29.807Reserved (RSVD1[0])—Offset 58h 3902
29.808Reserved (RSVD1[1])—Offset 5Ch 3902
29.809GPI Virtual Wire Message Enable (GPI_VWE_northwest_0)—Offset 60h 3903
29.810GPI Virtual Wire Message Enable (GPI_VWE_northwest_1)—Offset 64h 3911
29.811GPI Virtual Wire Message Enable (GPI_VWE_northwest_2)—Offset 68h 3920
29.812GPI Virtual Wire Message Enable (GPI_VWE_northwest_3)—Offset 6Ch 3928
29.813Reserved (RSVD2[0])—Offset 70h 3932
29.814Reserved (RSVD2[1])—Offset 74h 3932
29.815Reserved (RSVD2[2])—Offset 78h 3933
29.816Reserved (RSVD2[3])—Offset 7Ch 3933
29.817Pad Configuration Lock (PADCFGLOCK_northwest_0)—Offset 80h 3933
29.818Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_northwest_0)—Offset 84h 3949
29.819Pad Configuration Lock (PADCFGLOCK_northwest_1)—Offset 88h 3965
29.820Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_northwest_1)—Offset 8Ch 3981
29.821Pad Configuration Lock (PADCFGLOCK_northwest_2)—Offset 90h 3997
29.822Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_northwest_2)—Offset 94h 4013
29.823Pad Configuration Lock (PADCFGLOCK_northwest_3)—Offset 98h 4029
29.824Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_northwest_3)—Offset 9Ch 4037
29.825Reserved (RSVD3[0])—Offset A0h 4045
29.826Reserved (RSVD3[1])—Offset A4h 4045
29.827Reserved (RSVD3[2])—Offset A8h 4046
29.828Reserved (RSVD3[3])—Offset ACh 4046
29.829Host Software Pad Ownership (HOSTSW_OWN_northwest_0)—Offset B0h 4046
29.830Host Software Pad Ownership (HOSTSW_OWN_northwest_1)—Offset B4h 4053
29.831Host Software Pad Ownership (HOSTSW_OWN_northwest_2)—Offset B8h 4059
29.832Host Software Pad Ownership (HOSTSW_OWN_northwest_3)—Offset BCh 4066
29.833Reserved (RSVD4[0])—Offset C0h 4069
29.834Reserved (RSVD4[1])—Offset C4h 4070
29.835Reserved (RSVD4[2])—Offset C8h 4070
29.836Reserved (RSVD4[3])—Offset CCh 4070
29.837Reserved (RSVD4[4])—Offset D0h 4070
29.838Reserved (RSVD4[5])—Offset D4h 4071
29.839Reserved (RSVD4[6])—Offset D8h 4071
29.840Reserved (RSVD4[7])—Offset DCh 4071
29.841Reserved (RSVD4[8])—Offset E0h 4071
29.842Reserved (RSVD4[9])—Offset E4h 4072
29.843Reserved (RSVD4[10])—Offset E8h 4072
29.844Reserved (RSVD4[11])—Offset ECh 4072
29.845Reserved (RSVD4[12])—Offset F0h 4073
29.846Reserved (RSVD4[13])—Offset F4h 4073
29.847Reserved (RSVD4[14])—Offset F8h 4073
29.848Reserved (RSVD4[15])—Offset FCh 4073
29.849GPI Interrupt Status (GPI_IS_northwest_0)—Offset 100h 4074
29.850GPI Interrupt Status (GPI_IS_northwest_1)—Offset 104h 4082
29.851GPI Interrupt Status (GPI_IS_northwest_2)—Offset 108h 4090
29.852GPI Interrupt Status (GPI_IS_northwest_3)—Offset 10Ch 4098
29.853GPI Interrupt Enable (GPI_IE_northwest_0)—Offset 110h 4102
29.854GPI Interrupt Enable (GPI_IE_northwest_1)—Offset 114h 4108

29.855GPI Interrupt Enable (GPI_IE_northwest_2)—Offset 118h 4114
29.856GPI Interrupt Enable (GPI_IE_northwest_3)—Offset 11Ch 4119
29.857Reserved (RSVD5[0])—Offset 120h 4122
29.858Reserved (RSVD5[1])—Offset 124h 4122
29.859Reserved (RSVD5[2])—Offset 128h 4122
29.860Reserved (RSVD5[3])—Offset 12Ch 4123
29.861GPI General Purpose Events Status (GPI_GPE_STS_northwest_0)—Offset 130h 4123
29.862GPI General Purpose Events Status (GPI_GPE_STS_northwest_1)—Offset 134h 4131
29.863GPI General Purpose Events Status (GPI_GPE_STS_northwest_2)—Offset 138h 4140
29.864GPI General Purpose Events Status (GPI_GPE_STS_northwest_3)—Offset 13Ch 4148
29.865Reserved (RSVD6[0])—Offset 140h 4152
29.866Reserved (RSVD6[1])—Offset 144h 4152
29.867Reserved (RSVD6[2])—Offset 148h 4153
29.868Reserved (RSVD6[3])—Offset 14Ch 4153
29.869GPI General Purpose Events Enable (GPI_GPE_EN_northwest_0)—Offset 150h 4153
29.870GPI General Purpose Events Enable (GPI_GPE_EN_northwest_1)—Offset 154h 4160
29.871GPI General Purpose Events Enable (GPI_GPE_EN_northwest_2)—Offset 158h 4167
29.872GPI General Purpose Events Enable (GPI_GPE_EN_northwest_3)—Offset 15Ch 4174
29.873Reserved (RSVD7[0])—Offset 160h 4177
29.874Reserved (RSVD7[1])—Offset 164h 4177
29.875Reserved (RSVD7[2])—Offset 168h 4177
29.876Reserved (RSVD7[3])—Offset 16Ch 4178
29.877SMI Status (GPI_SMI_STS_northwest_0)—Offset 170h 4178
29.878SMI Status (GPI_SMI_STS_northwest_1)—Offset 174h 4189
29.879SMI Status (GPI_SMI_STS_northwest_2)—Offset 178h 4200
29.880SMI Status (GPI_SMI_STS_northwest_3)—Offset 17Ch 4211
29.881Reserved (RSVD8[0])—Offset 180h 4216
29.882Reserved (RSVD8[1])—Offset 184h 4216
29.883Reserved (RSVD8[2])—Offset 188h 4217
29.884Reserved (RSVD8[3])—Offset 18Ch 4217
29.885SMI Enable (GPI_SMI_EN_northwest_0)—Offset 190h 4217
29.886SMI Enable (GPI_SMI_EN_northwest_1)—Offset 194h 4225
29.887SMI Enable (GPI_SMI_EN_northwest_2)—Offset 198h 4233
29.888SMI Enable (GPI_SMI_EN_northwest_3)—Offset 19Ch 4241
29.889Reserved (RSVD9[0])—Offset 1A0h 4245
29.890Reserved (RSVD9[1])—Offset 1A4h 4246
29.891Reserved (RSVD9[2])—Offset 1A8h 4246
29.892Reserved (RSVD9[3])—Offset 1ACh 4246
29.893NMI Status (GPI_NMI_STS_northwest_0)—Offset 1B0h 4247
29.894NMI Status (GPI_NMI_STS_northwest_1)—Offset 1B4h 4255
29.895NMI Status (GPI_NMI_STS_northwest_2)—Offset 1B8h 4263
29.896NMI Status (GPI_NMI_STS_northwest_3)—Offset 1BCh 4271
29.897Reserved (RSVD10[0])—Offset 1C0h 4275
29.898Reserved (RSVD10[1])—Offset 1C4h 4276
29.899Reserved (RSVD10[2])—Offset 1C8h 4276
29.900Reserved (RSVD10[3])—Offset 1CCh 4276
29.901NMI Enable (GPI_NMI_EN_northwest_0)—Offset 1D0h 4276
29.902NMI Enable (GPI_NMI_EN_northwest_1)—Offset 1D4h 4284
29.903NMI Enable (GPI_NMI_EN_northwest_2)—Offset 1D8h 4292
29.904NMI Enable (GPI_NMI_EN_northwest_3)—Offset 1DCh 4300
29.905Reserved (RSVD11[0])—Offset 1E0h 4304
29.906Reserved (RSVD11[1])—Offset 1E4h 4304
29.907Reserved (RSVD11[2])—Offset 1E8h 4305
29.908Reserved (RSVD11[3])—Offset 1ECh 4305
29.909Reserved (RSVD11[4])—Offset 1F0h 4305

29.910Reserved (RSVD11[5])—Offset 1F4h 4306
29.911Reserved (RSVD11[6])—Offset 1F8h 4306
29.912Reserved (RSVD11[7])—Offset 1FCh 4306
29.913Reserved (RSVD11[8])—Offset 200h 4306
29.914Reserved (RSVD11[9])—Offset 204h 4307
29.915Reserved (RSVD11[10])—Offset 208h 4307
29.916Reserved (RSVD11[11])—Offset 20Ch 4307
29.917Event Trigger Output Enable (EVOUTEN_0)—Offset 210h 4307
29.918Event Trigger Output Enable (EVOUTEN_1)—Offset 214h 4308
29.919Event Trigger Mapping (EVMAP_0)—Offset 220h 4308
29.920Event Trigger Mapping (EVMAP_1)—Offset 228h 4311
29.921Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_JTAGITP_1P8)—Offset 300h 4315
29.922Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_JTAGITP_1P8)—Offset 304h 4315
29.923Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_JTAGITP_1P8)—Offset 308h 4316
29.924Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_JTAGITP_1P8)—Offset 30Ch 4316
29.925Family Configuration Register (FAM_CFG_JTAGITP_1P8)—Offset 310h 4317
29.926Family Reserved Register DW5 (FAM_RSVD_DW5JTAGITP_1P8)—Offset 314h 4318
29.927Family Reserved Register DW6 (FAM_RSVD_DW6JTAGITP_1P8)—Offset 318h 4319
29.928Family Reserved Register DW7 (FAM_RSVD_DW7JTAGITP_1P8)—Offset 31Ch 4319
29.929Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_GPHS1_1P8)—Offset 320h.. 4319
29.930Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_GPHS1_1P8)—Offset 324h.. 4320
29.931Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_GPHS1_1P8)—Offset 328h.. 4320
29.932Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_GPHS1_1P8)—Offset 32Ch . 4321
29.933Family Configuration Register (FAM_CFG_GPHS1_1P8)—Offset 330h 4321
29.934Family Reserved Register DW5 (FAM_RSVD_DW5GPHS1_1P8)—Offset 334h 4323
29.935Family Reserved Register DW6 (FAM_RSVD_DW6GPHS1_1P8)—Offset 338h 4323
29.936Family Reserved Register DW7 (FAM_RSVD_DW7GPHS1_1P8)—Offset 33Ch 4323
29.937Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_GPHS2_1P8)—Offset 340h.. 4324
29.938Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_GPHS2_1P8)—Offset 344h.. 4324
29.939Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_GPHS2_1P8)—Offset 348h.. 4325
29.940Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_GPHS2_1P8)—Offset 34Ch . 4325
29.941Family Configuration Register (FAM_CFG_GPHS2_1P8)—Offset 350h 4325
29.942Family Reserved Register DW5 (FAM_RSVD_DW5GPHS2_1P8)—Offset 354h 4327
29.943Family Reserved Register DW6 (FAM_RSVD_DW6GPHS2_1P8)—Offset 358h 4327
29.944Family Reserved Register DW7 (FAM_RSVD_DW7GPHS2_1P8)—Offset 35Ch 4328
29.945Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_MISC_1P8)—Offset 360h.... 4328
29.946Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_MISC_1P8)—Offset 364h.... 4328
29.947Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_MISC_1P8)—Offset 368h.... 4329
29.948Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_MISC_1P8)—Offset 36Ch.... 4329
29.949Family Configuration Register (FAM_CFG_MISC_1P8)—Offset 370h 4330
29.950Family Reserved Register DW5 (FAM_RSVD_DW5MISC_1P8)—Offset 374h 4331
29.951Family Reserved Register DW6 (FAM_RSVD_DW6MISC_1P8)—Offset 378h 4332
29.952Family Reserved Register DW7 (FAM_RSVD_DW7MISC_1P8)—Offset 37Ch 4332
29.953Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_I2C_1P8)—Offset 380h 4332
29.954Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_I2C_1P8)—Offset 384h 4333
29.955Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_I2C_1P8)—Offset 388h 4333
29.956Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_I2C_1P8)—Offset 38Ch 4334
29.957Family Configuration Register (FAM_CFG_I2C_1P8)—Offset 390h 4334
29.958Family Reserved Register DW5 (FAM_RSVD_DW5I2C_1P8)—Offset 394h 4336
29.959Family Reserved Register DW6 (FAM_RSVD_DW6I2C_1P8)—Offset 398h 4336
29.960Family Reserved Register DW7 (FAM_RSVD_DW7I2C_1P8)—Offset 39Ch 4336
29.961Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_UART_1P8)—Offset 3A0h ... 4337
29.962Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_UART_1P8)—Offset 3A4h ... 4337
29.963Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_UART_1P8)—Offset 3A8h ... 4338
29.964Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_UART_1P8)—Offset 3ACh ... 4338

29.965Family Configuration Register (FAM_CFG_UART_1P8)—Offset 3B0h 4338
29.966Family Reserved Register DW5 (FAM_RSVD_DW5UART_1P8)—Offset 3B4h.....4340
29.967Family Reserved Register DW6 (FAM_RSVD_DW6UART_1P8)—Offset 3B8h.....4340
29.968Family Reserved Register DW7 (FAM_RSVD_DW7UART_1P8)—Offset 3BCh 4341
29.969Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_PMC_1P8)—Offset 3C0h.....4341
29.970Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_PMC_1P8)—Offset 3C4h.....4341
29.971Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_PMC_1P8)—Offset 3C8h.....4342
29.972Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_PMC_1P8)—Offset 3CCh 4342
29.973Family Configuration Register (FAM_CFG_PMC_1P8)—Offset 3D0h 4343
29.974Family Reserved Register DW5 (FAM_RSVD_DW5PMC_1P8)—Offset 3D4h.....4344
29.975Family Reserved Register DW6 (FAM_RSVD_DW6PMC_1P8)—Offset 3D8h.....4345
29.976Family Reserved Register DW7 (FAM_RSVD_DW7PMC_1P8)—Offset 3DCh.....4345
29.977Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_v_CNV0)—Offset 3E0h.....4345
29.978Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_v_CNV0)—Offset 3E4h.....4346
29.979Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_v_CNV0)—Offset 3E8h.....4346
29.980Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_v_CNV0)—Offset 3ECh 4347
29.981Family Configuration Register (FAM_CFG_v_CNV0)—Offset 3F0h.....4347
29.982Family Reserved Register DW5 (FAM_RSVD_DW5v_CNV0)—Offset 3F4h 4349
29.983Family Reserved Register DW6 (FAM_RSVD_DW6v_CNV0)—Offset 3F8h 4349
29.984Family Reserved Register DW7 (FAM_RSVD_DW7v_CNV0)—Offset 3FCh 4349
29.985Pad Configuration DW0 (PAD_CFG_DW0_GPIO_0)—Offset 600h.....4350
29.986Pad Configuration DW1 (PAD_CFG_DW1_GPIO_0)—Offset 604h.....4352
29.987Pad Configuration DW2 (PAD_CFG_DW2_GPIO_0)—Offset 608h.....4354
29.988Pad Configuration DW3 (PAD_CFG_DW3_GPIO_0)—Offset 60Ch 4355
29.989Pad Configuration DW0 (PAD_CFG_DW0_GPIO_1)—Offset 610h.....4355
29.990Pad Configuration DW1 (PAD_CFG_DW1_GPIO_1)—Offset 614h.....4357
29.991Pad Configuration DW2 (PAD_CFG_DW2_GPIO_1)—Offset 618h.....4359
29.992Pad Configuration DW3 (PAD_CFG_DW3_GPIO_1)—Offset 61Ch 4360
29.993Pad Configuration DW0 (PAD_CFG_DW0_GPIO_2)—Offset 620h.....4360
29.994Pad Configuration DW1 (PAD_CFG_DW1_GPIO_2)—Offset 624h.....4362
29.995Pad Configuration DW2 (PAD_CFG_DW2_GPIO_2)—Offset 628h.....4364
29.996Pad Configuration DW3 (PAD_CFG_DW3_GPIO_2)—Offset 62Ch 4365
29.997Pad Configuration DW0 (PAD_CFG_DW0_GPIO_3)—Offset 630h.....4365
29.998Pad Configuration DW1 (PAD_CFG_DW1_GPIO_3)—Offset 634h.....4367
29.999Pad Configuration DW2 (PAD_CFG_DW2_GPIO_3)—Offset 638h.....4369
29.1000Pad Configuration DW3 (PAD_CFG_DW3_GPIO_3)—Offset 63Ch.....4370
29.1001Pad Configuration DW0 (PAD_CFG_DW0_GPIO_4)—Offset 640h.....4370
29.1002Pad Configuration DW1 (PAD_CFG_DW1_GPIO_4)—Offset 644h.....4372
29.1003Pad Configuration DW2 (PAD_CFG_DW2_GPIO_4)—Offset 648h.....4374
29.1004Pad Configuration DW3 (PAD_CFG_DW3_GPIO_4)—Offset 64Ch.....4375
29.1005Pad Configuration DW0 (PAD_CFG_DW0_GPIO_5)—Offset 650h.....4375
29.1006Pad Configuration DW1 (PAD_CFG_DW1_GPIO_5)—Offset 654h.....4377
29.1007Pad Configuration DW2 (PAD_CFG_DW2_GPIO_5)—Offset 658h.....4379
29.1008Pad Configuration DW3 (PAD_CFG_DW3_GPIO_5)—Offset 65Ch.....4380
29.1009Pad Configuration DW0 (PAD_CFG_DW0_GPIO_6)—Offset 660h.....4380
29.1010Pad Configuration DW1 (PAD_CFG_DW1_GPIO_6)—Offset 664h.....4382
29.1011Pad Configuration DW2 (PAD_CFG_DW2_GPIO_6)—Offset 668h.....4384
29.1012Pad Configuration DW3 (PAD_CFG_DW3_GPIO_6)—Offset 66Ch.....4385
29.1013Pad Configuration DW0 (PAD_CFG_DW0_GPIO_7)—Offset 670h.....4385
29.1014Pad Configuration DW1 (PAD_CFG_DW1_GPIO_7)—Offset 674h.....4387
29.1015Pad Configuration DW2 (PAD_CFG_DW2_GPIO_7)—Offset 678h.....4389
29.1016Pad Configuration DW3 (PAD_CFG_DW3_GPIO_7)—Offset 67Ch.....4390
29.1017Pad Configuration DW0 (PAD_CFG_DW0_GPIO_8)—Offset 680h.....4390
29.1018Pad Configuration DW1 (PAD_CFG_DW1_GPIO_8)—Offset 684h.....4392
29.1019Pad Configuration DW2 (PAD_CFG_DW2_GPIO_8)—Offset 688h.....4394

29.1020Pad Configuration DW3 (PAD_CFG_DW3_GPIO_8)—Offset 68Ch 4395
29.1021Pad Configuration DW0 (PAD_CFG_DW0_GPIO_9)—Offset 690h 4395
29.1022Pad Configuration DW1 (PAD_CFG_DW1_GPIO_9)—Offset 694h 4397
29.1023Pad Configuration DW2 (PAD_CFG_DW2_GPIO_9)—Offset 698h 4399
29.1024Pad Configuration DW3 (PAD_CFG_DW3_GPIO_9)—Offset 69Ch 4400
29.1025Pad Configuration DW0 (PAD_CFG_DW0_GPIO_10)—Offset 6A0h 4400
29.1026Pad Configuration DW1 (PAD_CFG_DW1_GPIO_10)—Offset 6A4h 4402
29.1027Pad Configuration DW2 (PAD_CFG_DW2_GPIO_10)—Offset 6A8h 4404
29.1028Pad Configuration DW3 (PAD_CFG_DW3_GPIO_10)—Offset 6ACh 4405
29.1029Pad Configuration DW0 (PAD_CFG_DW0_GPIO_11)—Offset 6B0h 4405
29.1030Pad Configuration DW1 (PAD_CFG_DW1_GPIO_11)—Offset 6B4h 4407
29.1031Pad Configuration DW2 (PAD_CFG_DW2_GPIO_11)—Offset 6B8h 4409
29.1032Pad Configuration DW3 (PAD_CFG_DW3_GPIO_11)—Offset 6BCh 4410
29.1033Pad Configuration DW0 (PAD_CFG_DW0_GPIO_12)—Offset 6C0h 4410
29.1034Pad Configuration DW1 (PAD_CFG_DW1_GPIO_12)—Offset 6C4h 4412
29.1035Pad Configuration DW2 (PAD_CFG_DW2_GPIO_12)—Offset 6C8h 4414
29.1036Pad Configuration DW3 (PAD_CFG_DW3_GPIO_12)—Offset 6CCh 4415
29.1037Pad Configuration DW0 (PAD_CFG_DW0_GPIO_13)—Offset 6D0h 4415
29.1038Pad Configuration DW1 (PAD_CFG_DW1_GPIO_13)—Offset 6D4h 4417
29.1039Pad Configuration DW2 (PAD_CFG_DW2_GPIO_13)—Offset 6D8h 4419
29.1040Pad Configuration DW3 (PAD_CFG_DW3_GPIO_13)—Offset 6DCh 4420
29.1041Pad Configuration DW0 (PAD_CFG_DW0_GPIO_14)—Offset 6E0h..... 4420
29.1042Pad Configuration DW1 (PAD_CFG_DW1_GPIO_14)—Offset 6E4h..... 4422
29.1043Pad Configuration DW2 (PAD_CFG_DW2_GPIO_14)—Offset 6E8h..... 4424
29.1044Pad Configuration DW3 (PAD_CFG_DW3_GPIO_14)—Offset 6ECh 4425
29.1045Pad Configuration DW0 (PAD_CFG_DW0_GPIO_15)—Offset 6F0h..... 4425
29.1046Pad Configuration DW1 (PAD_CFG_DW1_GPIO_15)—Offset 6F4h..... 4427
29.1047Pad Configuration DW2 (PAD_CFG_DW2_GPIO_15)—Offset 6F8h..... 4429
29.1048Pad Configuration DW3 (PAD_CFG_DW3_GPIO_15)—Offset 6FCh..... 4430
29.1049Pad Configuration DW0 (PAD_CFG_DW0_GPIO_16)—Offset 700h..... 4430
29.1050Pad Configuration DW1 (PAD_CFG_DW1_GPIO_16)—Offset 704h..... 4432
29.1051Pad Configuration DW2 (PAD_CFG_DW2_GPIO_16)—Offset 708h..... 4434
29.1052Pad Configuration DW3 (PAD_CFG_DW3_GPIO_16)—Offset 70Ch 4435
29.1053Pad Configuration DW0 (PAD_CFG_DW0_GPIO_17)—Offset 710h..... 4435
29.1054Pad Configuration DW1 (PAD_CFG_DW1_GPIO_17)—Offset 714h..... 4437
29.1055Pad Configuration DW2 (PAD_CFG_DW2_GPIO_17)—Offset 718h..... 4439
29.1056Pad Configuration DW3 (PAD_CFG_DW3_GPIO_17)—Offset 71Ch 4440
29.1057Pad Configuration DW0 (PAD_CFG_DW0_GPIO_18)—Offset 720h..... 4440
29.1058Pad Configuration DW1 (PAD_CFG_DW1_GPIO_18)—Offset 724h..... 4442
29.1059Pad Configuration DW2 (PAD_CFG_DW2_GPIO_18)—Offset 728h..... 4444
29.1060Pad Configuration DW3 (PAD_CFG_DW3_GPIO_18)—Offset 72Ch 4445
29.1061Pad Configuration DW0 (PAD_CFG_DW0_GPIO_19)—Offset 730h..... 4445
29.1062Pad Configuration DW1 (PAD_CFG_DW1_GPIO_19)—Offset 734h..... 4447
29.1063Pad Configuration DW2 (PAD_CFG_DW2_GPIO_19)—Offset 738h..... 4449
29.1064Pad Configuration DW3 (PAD_CFG_DW3_GPIO_19)—Offset 73Ch 4450
29.1065Pad Configuration DW0 (PAD_CFG_DW0_GPIO_20)—Offset 740h..... 4450
29.1066Pad Configuration DW1 (PAD_CFG_DW1_GPIO_20)—Offset 744h..... 4452
29.1067Pad Configuration DW2 (PAD_CFG_DW2_GPIO_20)—Offset 748h..... 4454
29.1068Pad Configuration DW3 (PAD_CFG_DW3_GPIO_20)—Offset 74Ch 4455
29.1069Pad Configuration DW0 (PAD_CFG_DW0_GPIO_21)—Offset 750h..... 4455
29.1070Pad Configuration DW1 (PAD_CFG_DW1_GPIO_21)—Offset 754h..... 4457
29.1071Pad Configuration DW2 (PAD_CFG_DW2_GPIO_21)—Offset 758h..... 4459
29.1072Pad Configuration DW3 (PAD_CFG_DW3_GPIO_21)—Offset 75Ch 4460
29.1073Pad Configuration DW0 (PAD_CFG_DW0_GPIO_22)—Offset 760h..... 4460
29.1074Pad Configuration DW1 (PAD_CFG_DW1_GPIO_22)—Offset 764h..... 4462

29.1075Pad Configuration DW2 (PAD_CFG_DW2_GPIO_22)—Offset 768h 4464
29.1076Pad Configuration DW3 (PAD_CFG_DW3_GPIO_22)—Offset 76Ch 4465
29.1077Pad Configuration DW0 (PAD_CFG_DW0_GPIO_23)—Offset 770h 4465
29.1078Pad Configuration DW1 (PAD_CFG_DW1_GPIO_23)—Offset 774h 4467
29.1079Pad Configuration DW2 (PAD_CFG_DW2_GPIO_23)—Offset 778h 4469
29.1080Pad Configuration DW3 (PAD_CFG_DW3_GPIO_23)—Offset 77Ch 4470
29.1081Pad Configuration DW0 (PAD_CFG_DW0_GPIO_24)—Offset 780h 4470
29.1082Pad Configuration DW1 (PAD_CFG_DW1_GPIO_24)—Offset 784h 4472
29.1083Pad Configuration DW2 (PAD_CFG_DW2_GPIO_24)—Offset 788h 4474
29.1084Pad Configuration DW3 (PAD_CFG_DW3_GPIO_24)—Offset 78Ch 4475
29.1085Pad Configuration DW0 (PAD_CFG_DW0_GPIO_25)—Offset 790h 4475
29.1086Pad Configuration DW1 (PAD_CFG_DW1_GPIO_25)—Offset 794h 4477
29.1087Pad Configuration DW2 (PAD_CFG_DW2_GPIO_25)—Offset 798h 4479
29.1088Pad Configuration DW3 (PAD_CFG_DW3_GPIO_25)—Offset 79Ch 4480
29.1089Pad Configuration DW0 (PAD_CFG_DW0_GPIO_26)—Offset 7A0h 4480
29.1090Pad Configuration DW1 (PAD_CFG_DW1_GPIO_26)—Offset 7A4h 4482
29.1091Pad Configuration DW2 (PAD_CFG_DW2_GPIO_26)—Offset 7A8h 4484
29.1092Pad Configuration DW3 (PAD_CFG_DW3_GPIO_26)—Offset 7ACh 4485
29.1093Pad Configuration DW0 (PAD_CFG_DW0_GPIO_27)—Offset 7B0h 4485
29.1094Pad Configuration DW1 (PAD_CFG_DW1_GPIO_27)—Offset 7B4h 4487
29.1095Pad Configuration DW2 (PAD_CFG_DW2_GPIO_27)—Offset 7B8h 4489
29.1096Pad Configuration DW3 (PAD_CFG_DW3_GPIO_27)—Offset 7BCh 4490
29.1097Pad Configuration DW0 (PAD_CFG_DW0_GPIO_28)—Offset 7C0h 4490
29.1098Pad Configuration DW1 (PAD_CFG_DW1_GPIO_28)—Offset 7C4h 4492
29.1099Pad Configuration DW2 (PAD_CFG_DW2_GPIO_28)—Offset 7C8h 4494
29.1100Pad Configuration DW3 (PAD_CFG_DW3_GPIO_28)—Offset 7CCh 4495
29.1101Pad Configuration DW0 (PAD_CFG_DW0_GPIO_29)—Offset 7D0h 4495
29.1102Pad Configuration DW1 (PAD_CFG_DW1_GPIO_29)—Offset 7D4h 4497
29.1103Pad Configuration DW2 (PAD_CFG_DW2_GPIO_29)—Offset 7D8h 4499
29.1104Pad Configuration DW3 (PAD_CFG_DW3_GPIO_29)—Offset 7DCh 4500
29.1105Pad Configuration DW0 (PAD_CFG_DW0_GPIO_30)—Offset 7E0h 4500
29.1106Pad Configuration DW1 (PAD_CFG_DW1_GPIO_30)—Offset 7E4h 4502
29.1107Pad Configuration DW2 (PAD_CFG_DW2_GPIO_30)—Offset 7E8h 4504
29.1108Pad Configuration DW3 (PAD_CFG_DW3_GPIO_30)—Offset 7ECh 4505
29.1109Pad Configuration DW0 (PAD_CFG_DW0_GPIO_31)—Offset 7F0h 4505
29.1110Pad Configuration DW1 (PAD_CFG_DW1_GPIO_31)—Offset 7F4h 4507
29.1111Pad Configuration DW2 (PAD_CFG_DW2_GPIO_31)—Offset 7F8h 4509
29.1112Pad Configuration DW3 (PAD_CFG_DW3_GPIO_31)—Offset 7FCh 4510
29.1113Pad Configuration DW0 (PAD_CFG_DW0_GPIO_32)—Offset 800h 4510
29.1114Pad Configuration DW1 (PAD_CFG_DW1_GPIO_32)—Offset 804h 4512
29.1115Pad Configuration DW2 (PAD_CFG_DW2_GPIO_32)—Offset 808h 4514
29.1116Pad Configuration DW3 (PAD_CFG_DW3_GPIO_32)—Offset 80Ch 4515
29.1117Pad Configuration DW0 (PAD_CFG_DW0_GPIO_33)—Offset 810h 4515
29.1118Pad Configuration DW1 (PAD_CFG_DW1_GPIO_33)—Offset 814h 4517
29.1119Pad Configuration DW2 (PAD_CFG_DW2_GPIO_33)—Offset 818h 4519
29.1120Pad Configuration DW3 (PAD_CFG_DW3_GPIO_33)—Offset 81Ch 4520
29.1121Pad Configuration DW0 (PAD_CFG_DW0_GPIO_34)—Offset 820h 4520
29.1122Pad Configuration DW1 (PAD_CFG_DW1_GPIO_34)—Offset 824h 4522
29.1123Pad Configuration DW2 (PAD_CFG_DW2_GPIO_34)—Offset 828h 4524
29.1124Pad Configuration DW3 (PAD_CFG_DW3_GPIO_34)—Offset 82Ch 4525
29.1125Pad Configuration DW0 (PAD_CFG_DW0_GPIO_35)—Offset 830h 4525
29.1126Pad Configuration DW1 (PAD_CFG_DW1_GPIO_35)—Offset 834h 4527
29.1127Pad Configuration DW2 (PAD_CFG_DW2_GPIO_35)—Offset 838h 4529
29.1128Pad Configuration DW3 (PAD_CFG_DW3_GPIO_35)—Offset 83Ch 4530
29.1129Pad Configuration DW0 (PAD_CFG_DW0_GPIO_36)—Offset 840h 4530

29.1130Pad Configuration DW1 (PAD_CFG_DW1_GPIO_36)—Offset 844h..... 4532
29.1131Pad Configuration DW2 (PAD_CFG_DW2_GPIO_36)—Offset 848h..... 4534
29.1132Pad Configuration DW3 (PAD_CFG_DW3_GPIO_36)—Offset 84Ch 4535
29.1133Pad Configuration DW0 (PAD_CFG_DW0_GPIO_37)—Offset 850h..... 4535
29.1134Pad Configuration DW1 (PAD_CFG_DW1_GPIO_37)—Offset 854h..... 4537
29.1135Pad Configuration DW2 (PAD_CFG_DW2_GPIO_37)—Offset 858h..... 4539
29.1136Pad Configuration DW3 (PAD_CFG_DW3_GPIO_37)—Offset 85Ch 4540
29.1137Pad Configuration DW0 (PAD_CFG_DW0_GPIO_38)—Offset 860h..... 4540
29.1138Pad Configuration DW1 (PAD_CFG_DW1_GPIO_38)—Offset 864h..... 4542
29.1139Pad Configuration DW2 (PAD_CFG_DW2_GPIO_38)—Offset 868h..... 4544
29.1140Pad Configuration DW3 (PAD_CFG_DW3_GPIO_38)—Offset 86Ch 4545
29.1141Pad Configuration DW0 (PAD_CFG_DW0_GPIO_39)—Offset 870h..... 4545
29.1142Pad Configuration DW1 (PAD_CFG_DW1_GPIO_39)—Offset 874h..... 4547
29.1143Pad Configuration DW2 (PAD_CFG_DW2_GPIO_39)—Offset 878h..... 4549
29.1144Pad Configuration DW3 (PAD_CFG_DW3_GPIO_39)—Offset 87Ch 4550
29.1145Pad Configuration DW0 (PAD_CFG_DW0_GPIO_40)—Offset 880h..... 4550
29.1146Pad Configuration DW1 (PAD_CFG_DW1_GPIO_40)—Offset 884h..... 4552
29.1147Pad Configuration DW2 (PAD_CFG_DW2_GPIO_40)—Offset 888h..... 4554
29.1148Pad Configuration DW3 (PAD_CFG_DW3_GPIO_40)—Offset 88Ch 4555
29.1149Pad Configuration DW0 (PAD_CFG_DW0_GPIO_41)—Offset 890h..... 4555
29.1150Pad Configuration DW1 (PAD_CFG_DW1_GPIO_41)—Offset 894h..... 4557
29.1151Pad Configuration DW2 (PAD_CFG_DW2_GPIO_41)—Offset 898h..... 4559
29.1152Pad Configuration DW3 (PAD_CFG_DW3_GPIO_41)—Offset 89Ch 4560
29.1153Pad Configuration DW0 (PAD_CFG_DW0_GPIO_42)—Offset 8A0h 4560
29.1154Pad Configuration DW1 (PAD_CFG_DW1_GPIO_42)—Offset 8A4h 4562
29.1155Pad Configuration DW2 (PAD_CFG_DW2_GPIO_42)—Offset 8A8h 4564
29.1156Pad Configuration DW3 (PAD_CFG_DW3_GPIO_42)—Offset 8ACh 4565
29.1157Pad Configuration DW0 (PAD_CFG_DW0_GPIO_43)—Offset 8B0h 4565
29.1158Pad Configuration DW1 (PAD_CFG_DW1_GPIO_43)—Offset 8B4h 4567
29.1159Pad Configuration DW2 (PAD_CFG_DW2_GPIO_43)—Offset 8B8h 4569
29.1160Pad Configuration DW3 (PAD_CFG_DW3_GPIO_43)—Offset 8BCh 4570
29.1161Pad Configuration DW0 (PAD_CFG_DW0_GPIO_44)—Offset 8C0h 4570
29.1162Pad Configuration DW1 (PAD_CFG_DW1_GPIO_44)—Offset 8C4h 4572
29.1163Pad Configuration DW2 (PAD_CFG_DW2_GPIO_44)—Offset 8C8h 4574
29.1164Pad Configuration DW3 (PAD_CFG_DW3_GPIO_44)—Offset 8CCh 4575
29.1165Pad Configuration DW0 (PAD_CFG_DW0_GPIO_45)—Offset 8D0h 4575
29.1166Pad Configuration DW1 (PAD_CFG_DW1_GPIO_45)—Offset 8D4h 4577
29.1167Pad Configuration DW2 (PAD_CFG_DW2_GPIO_45)—Offset 8D8h 4579
29.1168Pad Configuration DW3 (PAD_CFG_DW3_GPIO_45)—Offset 8DCh 4580
29.1169Pad Configuration DW0 (PAD_CFG_DW0_GPIO_46)—Offset 8E0h..... 4580
29.1170Pad Configuration DW1 (PAD_CFG_DW1_GPIO_46)—Offset 8E4h..... 4582
29.1171Pad Configuration DW2 (PAD_CFG_DW2_GPIO_46)—Offset 8E8h..... 4584
29.1172Pad Configuration DW3 (PAD_CFG_DW3_GPIO_46)—Offset 8ECh 4585
29.1173Pad Configuration DW0 (PAD_CFG_DW0_GPIO_47)—Offset 8F0h..... 4585
29.1174Pad Configuration DW1 (PAD_CFG_DW1_GPIO_47)—Offset 8F4h..... 4587
29.1175Pad Configuration DW2 (PAD_CFG_DW2_GPIO_47)—Offset 8F8h..... 4589
29.1176Pad Configuration DW3 (PAD_CFG_DW3_GPIO_47)—Offset 8FCh..... 4590
29.1177Pad Configuration DW0 (PAD_CFG_DW0_GPIO_48)—Offset 900h..... 4590
29.1178Pad Configuration DW1 (PAD_CFG_DW1_GPIO_48)—Offset 904h..... 4592
29.1179Pad Configuration DW2 (PAD_CFG_DW2_GPIO_48)—Offset 908h..... 4594
29.1180Pad Configuration DW3 (PAD_CFG_DW3_GPIO_48)—Offset 90Ch 4595
29.1181Pad Configuration DW0 (PAD_CFG_DW0_GPIO_49)—Offset 910h..... 4595
29.1182Pad Configuration DW1 (PAD_CFG_DW1_GPIO_49)—Offset 914h..... 4597
29.1183Pad Configuration DW2 (PAD_CFG_DW2_GPIO_49)—Offset 918h..... 4599
29.1184Pad Configuration DW3 (PAD_CFG_DW3_GPIO_49)—Offset 91Ch 4600

29.1185Pad Configuration DW0 (PAD_CFG_DW0_GPIO_50)—Offset 920h 4600
29.1186Pad Configuration DW1 (PAD_CFG_DW1_GPIO_50)—Offset 924h 4602
29.1187Pad Configuration DW2 (PAD_CFG_DW2_GPIO_50)—Offset 928h 4604
29.1188Pad Configuration DW3 (PAD_CFG_DW3_GPIO_50)—Offset 92Ch 4605
29.1189Pad Configuration DW0 (PAD_CFG_DW0_GPIO_51)—Offset 930h 4605
29.1190Pad Configuration DW1 (PAD_CFG_DW1_GPIO_51)—Offset 934h 4607
29.1191Pad Configuration DW2 (PAD_CFG_DW2_GPIO_51)—Offset 938h 4609
29.1192Pad Configuration DW3 (PAD_CFG_DW3_GPIO_51)—Offset 93Ch 4610
29.1193Pad Configuration DW0 (PAD_CFG_DW0_GPIO_52)—Offset 940h 4610
29.1194Pad Configuration DW1 (PAD_CFG_DW1_GPIO_52)—Offset 944h 4612
29.1195Pad Configuration DW2 (PAD_CFG_DW2_GPIO_52)—Offset 948h 4614
29.1196Pad Configuration DW3 (PAD_CFG_DW3_GPIO_52)—Offset 94Ch 4615
29.1197Pad Configuration DW0 (PAD_CFG_DW0_GPIO_53)—Offset 950h 4615
29.1198Pad Configuration DW1 (PAD_CFG_DW1_GPIO_53)—Offset 954h 4617
29.1199Pad Configuration DW2 (PAD_CFG_DW2_GPIO_53)—Offset 958h 4619
29.1200Pad Configuration DW3 (PAD_CFG_DW3_GPIO_53)—Offset 95Ch 4620
29.1201Pad Configuration DW0 (PAD_CFG_DW0_GPIO_54)—Offset 960h 4620
29.1202Pad Configuration DW1 (PAD_CFG_DW1_GPIO_54)—Offset 964h 4622
29.1203Pad Configuration DW2 (PAD_CFG_DW2_GPIO_54)—Offset 968h 4624
29.1204Pad Configuration DW3 (PAD_CFG_DW3_GPIO_54)—Offset 96Ch 4625
29.1205Pad Configuration DW0 (PAD_CFG_DW0_GPIO_55)—Offset 970h 4625
29.1206Pad Configuration DW1 (PAD_CFG_DW1_GPIO_55)—Offset 974h 4627
29.1207Pad Configuration DW2 (PAD_CFG_DW2_GPIO_55)—Offset 978h 4629
29.1208Pad Configuration DW3 (PAD_CFG_DW3_GPIO_55)—Offset 97Ch 4630
29.1209Pad Configuration DW0 (PAD_CFG_DW0_GPIO_56)—Offset 980h 4630
29.1210Pad Configuration DW1 (PAD_CFG_DW1_GPIO_56)—Offset 984h 4632
29.1211Pad Configuration DW2 (PAD_CFG_DW2_GPIO_56)—Offset 988h 4634
29.1212Pad Configuration DW3 (PAD_CFG_DW3_GPIO_56)—Offset 98Ch 4635
29.1213Pad Configuration DW0 (PAD_CFG_DW0_GPIO_57)—Offset 990h 4635
29.1214Pad Configuration DW1 (PAD_CFG_DW1_GPIO_57)—Offset 994h 4637
29.1215Pad Configuration DW2 (PAD_CFG_DW2_GPIO_57)—Offset 998h 4639
29.1216Pad Configuration DW3 (PAD_CFG_DW3_GPIO_57)—Offset 99Ch 4640
29.1217Pad Configuration DW0 (PAD_CFG_DW0_GPIO_58)—Offset 9A0h 4640
29.1218Pad Configuration DW1 (PAD_CFG_DW1_GPIO_58)—Offset 9A4h 4642
29.1219Pad Configuration DW2 (PAD_CFG_DW2_GPIO_58)—Offset 9A8h 4644
29.1220Pad Configuration DW3 (PAD_CFG_DW3_GPIO_58)—Offset 9ACh 4645
29.1221Pad Configuration DW0 (PAD_CFG_DW0_GPIO_59)—Offset 9B0h 4645
29.1222Pad Configuration DW1 (PAD_CFG_DW1_GPIO_59)—Offset 9B4h 4647
29.1223Pad Configuration DW2 (PAD_CFG_DW2_GPIO_59)—Offset 9B8h 4649
29.1224Pad Configuration DW3 (PAD_CFG_DW3_GPIO_59)—Offset 9BCh 4650
29.1225Pad Configuration DW0 (PAD_CFG_DW0_GPIO_60)—Offset 9C0h 4650
29.1226Pad Configuration DW1 (PAD_CFG_DW1_GPIO_60)—Offset 9C4h 4652
29.1227Pad Configuration DW2 (PAD_CFG_DW2_GPIO_60)—Offset 9C8h 4654
29.1228Pad Configuration DW3 (PAD_CFG_DW3_GPIO_60)—Offset 9CCh 4655
29.1229Pad Configuration DW0 (PAD_CFG_DW0_GPIO_61)—Offset 9D0h 4655
29.1230Pad Configuration DW1 (PAD_CFG_DW1_GPIO_61)—Offset 9D4h 4657
29.1231Pad Configuration DW2 (PAD_CFG_DW2_GPIO_61)—Offset 9D8h 4659
29.1232Pad Configuration DW3 (PAD_CFG_DW3_GPIO_61)—Offset 9DCh 4660
29.1233Pad Configuration DW0 (PAD_CFG_DW0_GPIO_62)—Offset 9E0h 4660
29.1234Pad Configuration DW1 (PAD_CFG_DW1_GPIO_62)—Offset 9E4h 4662
29.1235Pad Configuration DW2 (PAD_CFG_DW2_GPIO_62)—Offset 9E8h 4664
29.1236Pad Configuration DW3 (PAD_CFG_DW3_GPIO_62)—Offset 9ECh 4665
29.1237Pad Configuration DW0 (PAD_CFG_DW0_GPIO_63)—Offset 9F0h 4665
29.1238Pad Configuration DW1 (PAD_CFG_DW1_GPIO_63)—Offset 9F4h 4667
29.1239Pad Configuration DW2 (PAD_CFG_DW2_GPIO_63)—Offset 9F8h 4669

29.1240Pad Configuration DW3 (PAD_CFG_DW3_GPIO_63)—Offset 9FCh..... 4670
29.1241Pad Configuration DW0 (PAD_CFG_DW0_GPIO_64)—Offset A00h 4670
29.1242Pad Configuration DW1 (PAD_CFG_DW1_GPIO_64)—Offset A04h 4672
29.1243Pad Configuration DW2 (PAD_CFG_DW2_GPIO_64)—Offset A08h 4674
29.1244Pad Configuration DW3 (PAD_CFG_DW3_GPIO_64)—Offset A0Ch 4675
29.1245Pad Configuration DW0 (PAD_CFG_DW0_GPIO_65)—Offset A10h 4675
29.1246Pad Configuration DW1 (PAD_CFG_DW1_GPIO_65)—Offset A14h 4677
29.1247Pad Configuration DW2 (PAD_CFG_DW2_GPIO_65)—Offset A18h 4679
29.1248Pad Configuration DW3 (PAD_CFG_DW3_GPIO_65)—Offset A1Ch 4680
29.1249Pad Configuration DW0 (PAD_CFG_DW0_GPIO_66)—Offset A20h 4680
29.1250Pad Configuration DW1 (PAD_CFG_DW1_GPIO_66)—Offset A24h 4682
29.1251Pad Configuration DW2 (PAD_CFG_DW2_GPIO_66)—Offset A28h 4684
29.1252Pad Configuration DW3 (PAD_CFG_DW3_GPIO_66)—Offset A2Ch 4685
29.1253Pad Configuration DW0 (PAD_CFG_DW0_GPIO_67)—Offset A30h 4685
29.1254Pad Configuration DW1 (PAD_CFG_DW1_GPIO_67)—Offset A34h 4687
29.1255Pad Configuration DW2 (PAD_CFG_DW2_GPIO_67)—Offset A38h 4689
29.1256Pad Configuration DW3 (PAD_CFG_DW3_GPIO_67)—Offset A3Ch 4690
29.1257Pad Configuration DW0 (PAD_CFG_DW0_GPIO_68)—Offset A40h 4690
29.1258Pad Configuration DW1 (PAD_CFG_DW1_GPIO_68)—Offset A44h 4692
29.1259Pad Configuration DW2 (PAD_CFG_DW2_GPIO_68)—Offset A48h 4694
29.1260Pad Configuration DW3 (PAD_CFG_DW3_GPIO_68)—Offset A4Ch 4695
29.1261Pad Configuration DW0 (PAD_CFG_DW0_GPIO_69)—Offset A50h 4695
29.1262Pad Configuration DW1 (PAD_CFG_DW1_GPIO_69)—Offset A54h 4697
29.1263Pad Configuration DW2 (PAD_CFG_DW2_GPIO_69)—Offset A58h 4699
29.1264Pad Configuration DW3 (PAD_CFG_DW3_GPIO_69)—Offset A5Ch 4700
29.1265Pad Configuration DW0 (PAD_CFG_DW0_GPIO_70)—Offset A60h 4700
29.1266Pad Configuration DW1 (PAD_CFG_DW1_GPIO_70)—Offset A64h 4702
29.1267Pad Configuration DW2 (PAD_CFG_DW2_GPIO_70)—Offset A68h 4704
29.1268Pad Configuration DW3 (PAD_CFG_DW3_GPIO_70)—Offset A6Ch 4705
29.1269Pad Configuration DW0 (PAD_CFG_DW0_GPIO_71)—Offset A70h 4705
29.1270Pad Configuration DW1 (PAD_CFG_DW1_GPIO_71)—Offset A74h 4707
29.1271Pad Configuration DW2 (PAD_CFG_DW2_GPIO_71)—Offset A78h 4709
29.1272Pad Configuration DW3 (PAD_CFG_DW3_GPIO_71)—Offset A7Ch 4710
29.1273Pad Configuration DW0 (PAD_CFG_DW0_GPIO_72)—Offset A80h 4710
29.1274Pad Configuration DW1 (PAD_CFG_DW1_GPIO_72)—Offset A84h 4712
29.1275Pad Configuration DW2 (PAD_CFG_DW2_GPIO_72)—Offset A88h 4714
29.1276Pad Configuration DW3 (PAD_CFG_DW3_GPIO_72)—Offset A8Ch 4715
29.1277Pad Configuration DW0 (PAD_CFG_DW0_GPIO_73)—Offset A90h 4715
29.1278Pad Configuration DW1 (PAD_CFG_DW1_GPIO_73)—Offset A94h 4717
29.1279Pad Configuration DW2 (PAD_CFG_DW2_GPIO_73)—Offset A98h 4719
29.1280Pad Configuration DW3 (PAD_CFG_DW3_GPIO_73)—Offset A9Ch 4720
29.1281Pad Configuration DW0 (PAD_CFG_DW0_GPIO_74)—Offset AA0h 4720
29.1282Pad Configuration DW1 (PAD_CFG_DW1_GPIO_74)—Offset AA4h 4722
29.1283Pad Configuration DW2 (PAD_CFG_DW2_GPIO_74)—Offset AA8h 4724
29.1284Pad Configuration DW3 (PAD_CFG_DW3_GPIO_74)—Offset AACh 4725
29.1285Pad Configuration DW0 (PAD_CFG_DW0_GPIO_75)—Offset AB0h 4725
29.1286Pad Configuration DW1 (PAD_CFG_DW1_GPIO_75)—Offset AB4h 4727
29.1287Pad Configuration DW2 (PAD_CFG_DW2_GPIO_75)—Offset AB8h 4729
29.1288Pad Configuration DW3 (PAD_CFG_DW3_GPIO_75)—Offset ABCh 4730
29.1289Pad Configuration DW0 (PAD_CFG_DW0_GPIO_211)—Offset AC0h 4730
29.1290Pad Configuration DW1 (PAD_CFG_DW1_GPIO_211)—Offset AC4h 4732
29.1291Pad Configuration DW2 (PAD_CFG_DW2_GPIO_211)—Offset AC8h 4734
29.1292Pad Configuration DW3 (PAD_CFG_DW3_GPIO_211)—Offset ACCh 4735
29.1293Pad Configuration DW0 (PAD_CFG_DW0_GPIO_212)—Offset AD0h 4735
29.1294Pad Configuration DW1 (PAD_CFG_DW1_GPIO_212)—Offset AD4h 4737

29.1295Pad Configuration DW2 (PAD_CFG_DW2_GPIO_212)—Offset AD8h 4739
29.1296Pad Configuration DW3 (PAD_CFG_DW3_GPIO_212)—Offset ADCh 4740
29.1297Pad Configuration DW0 (PAD_CFG_DW0_GPIO_213)—Offset AE0h 4740
29.1298Pad Configuration DW1 (PAD_CFG_DW1_GPIO_213)—Offset AE4h 4742
29.1299Pad Configuration DW2 (PAD_CFG_DW2_GPIO_213)—Offset AE8h 4744
29.1300Pad Configuration DW3 (PAD_CFG_DW3_GPIO_213)—Offset AECh 4745
29.1301Pad Configuration DW0 (PAD_CFG_DW0_GPIO_214)—Offset AF0h 4745
29.1302Pad Configuration DW1 (PAD_CFG_DW1_GPIO_214)—Offset AF4h 4747
29.1303Pad Configuration DW2 (PAD_CFG_DW2_GPIO_214)—Offset AF8h 4749
29.1304Pad Configuration DW3 (PAD_CFG_DW3_GPIO_214)—Offset AFCh 4750
29.1305Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_0)—Offset B00h 4750
29.1306Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_0)—Offset B04h 4752
29.1307Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_0)—Offset B08h 4754
29.1308Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_0)—Offset B0Ch 4755
29.1309Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_1)—Offset B10h 4755
29.1310Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_1)—Offset B14h 4757
29.1311Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_1)—Offset B18h 4759
29.1312Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_1)—Offset B1Ch 4760
29.1313Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_2)—Offset B20h 4760
29.1314Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_2)—Offset B24h 4762
29.1315Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_2)—Offset B28h 4764
29.1316Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_2)—Offset B2Ch 4765
29.1317Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_3)—Offset B30h 4765
29.1318Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_3)—Offset B34h 4767
29.1319Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_3)—Offset B38h 4769
29.1320Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_3)—Offset B3Ch 4770
29.1321Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_4)—Offset B40h 4770
29.1322Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_4)—Offset B44h 4772
29.1323Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_4)—Offset B48h 4774
29.1324Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_4)—Offset B4Ch 4775
29.1325Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_5)—Offset B50h 4775
29.1326Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_5)—Offset B54h 4777
29.1327Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_5)—Offset B58h 4779
29.1328Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_5)—Offset B5Ch 4780
29.1329Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_6)—Offset B60h 4780
29.1330Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_6)—Offset B64h 4782
29.1331Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_6)—Offset B68h 4784
29.1332Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_6)—Offset B6Ch 4785
29.1333Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_7)—Offset B70h 4785
29.1334Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_7)—Offset B74h 4787
29.1335Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_7)—Offset B78h 4789
29.1336Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_7)—Offset B7Ch 4790
29.1337Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_8)—Offset B80h 4790
29.1338Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_8)—Offset B84h 4792
29.1339Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_8)—Offset B88h 4794
29.1340Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_8)—Offset B8Ch 4795
29.1341Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_9)—Offset B90h 4795
29.1342Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_9)—Offset B94h 4797
29.1343Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_9)—Offset B98h 4799
29.1344Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_9)—Offset B9Ch 4800
29.1345Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_10)—Offset BA0h 4800
29.1346Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_10)—Offset BA4h 4802
29.1347Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_10)—Offset BA8h 4804
29.1348Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_10)—Offset BACh 4805
29.1349Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_11)—Offset BB0h 4805

29.1350Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_11)—Offset BB4h 4807
29.1351Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_11)—Offset BB8h 4809
29.1352Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_11)—Offset BBCh 4810
29.1353Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_12)—Offset BC0h 4810
29.1354Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_12)—Offset BC4h 4812
29.1355Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_12)—Offset BC8h 4814
29.1356Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_12)—Offset BCCh 4815
29.1357Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_13)—Offset BD0h 4815
29.1358Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_13)—Offset BD4h 4817
29.1359Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_13)—Offset BD8h 4819
29.1360Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_13)—Offset BDCh 4820
29.1361Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_14)—Offset BE0h 4820
29.1362Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_14)—Offset BE4h 4822
29.1363Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_14)—Offset BE8h 4824
29.1364Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_14)—Offset BECh 4825
29.1365Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_15)—Offset BF0h 4825
29.1366Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_15)—Offset BF4h 4827
29.1367Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_15)—Offset BF8h 4829
29.1368Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_15)—Offset BFCh 4830
29.1369Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_16)—Offset C00h 4830
29.1370Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_16)—Offset C04h 4832
29.1371Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_16)—Offset C08h 4834
29.1372Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_16)—Offset C0Ch 4835
29.1373Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_17)—Offset C10h 4835
29.1374Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_17)—Offset C14h 4837
29.1375Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_17)—Offset C18h 4839
29.1376Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_17)—Offset C1Ch 4840
29.1377Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_18)—Offset C20h 4840
29.1378Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_18)—Offset C24h 4842
29.1379Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_18)—Offset C28h 4844
29.1380Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_18)—Offset C2Ch 4845
29.1381Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_19)—Offset C30h 4845
29.1382Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_19)—Offset C34h 4847
29.1383Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_19)—Offset C38h 4849
29.1384Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_19)—Offset C3Ch 4850
29.1385Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_20)—Offset C40h 4850
29.1386Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_20)—Offset C44h 4852
29.1387Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_20)—Offset C48h 4854
29.1388Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_20)—Offset C4Ch 4855
29.1389Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_21)—Offset C50h 4855
29.1390Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_21)—Offset C54h 4857
29.1391Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_21)—Offset C58h 4859
29.1392Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_21)—Offset C5Ch 4860
29.1393Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_22)—Offset C60h 4860
29.1394Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_22)—Offset C64h 4862
29.1395Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_22)—Offset C68h 4864
29.1396Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_22)—Offset C6Ch 4865
29.1397Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_23)—Offset C70h 4865
29.1398Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_23)—Offset C74h 4867
29.1399Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_23)—Offset C78h 4869
29.1400Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_23)—Offset C7Ch 4870
29.1401Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_24)—Offset C80h 4870
29.1402Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_24)—Offset C84h 4872
29.1403Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_24)—Offset C88h 4874
29.1404Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_24)—Offset C8Ch 4875

29.1405Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_25)—Offset C90h 4875
29.1406Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_25)—Offset C94h 4877
29.1407Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_25)—Offset C98h 4879
29.1408Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_25)—Offset C9Ch 4880
29.1409Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_26)—Offset CA0h 4880
29.1410Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_26)—Offset CA4h 4882
29.1411Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_26)—Offset CA8h 4884
29.1412Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_26)—Offset CACh 4885
29.1413Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_27)—Offset CB0h 4885
29.1414Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_27)—Offset CB4h 4887
29.1415Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_27)—Offset CB8h 4889
29.1416Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_27)—Offset CBCh 4890
29.1417Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_28)—Offset CC0h 4890
29.1418Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_28)—Offset CC4h 4892
29.1419Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_28)—Offset CC8h 4894
29.1420Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_28)—Offset CCCh 4895
29.1421Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_29)—Offset CD0h 4895
29.1422Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_29)—Offset CD4h 4897
29.1423Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_29)—Offset CD8h 4899
29.1424Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_29)—Offset CDCh 4900
29.1425Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_30)—Offset CE0h 4900
29.1426Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_30)—Offset CE4h 4902
29.1427Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_30)—Offset CE8h 4904
29.1428Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_30)—Offset CECh 4905
29.1429Capability List Register (CAP_LIST_0)—Offset 4h 4906
29.1430Family Base Address (FAMBAR)—Offset 8h 4906
29.1431Pad Base Address (PADBAR)—Offset Ch 4906
29.1432Miscellaneous Configuration (MISCCFG)—Offset 10h 4907
29.1433Miscellaneous Secured Configuration (MISCSECCFG)—Offset 14h 4908
29.1434Reserved (RSVD0[0])—Offset 18h 4908
29.1435Reserved (RSVD0[1])—Offset 1Ch 4909
29.1436Pad Ownership (PAD_OWN_scc_0)—Offset 20h 4909
29.1437Pad Ownership (PAD_OWN_scc_1)—Offset 24h 4914
29.1438Pad Ownership (PAD_OWN_scc_2)—Offset 28h 4918
29.1439Pad Ownership (PAD_OWN_scc_3)—Offset 2Ch 4923
29.1440Pad Ownership (PAD_OWN_scc_4)—Offset 30h 4927
29.1441Reserved (RSVD1[0])—Offset 34h 4929
29.1442Reserved (RSVD1[1])—Offset 38h 4929
29.1443Reserved (RSVD1[2])—Offset 3Ch 4930
29.1444Reserved (RSVD1[3])—Offset 40h 4930
29.1445Reserved (RSVD1[4])—Offset 44h 4930
29.1446Reserved (RSVD1[5])—Offset 48h 4930
29.1447Reserved (RSVD1[6])—Offset 4Ch 4931
29.1448Reserved (RSVD1[7])—Offset 50h 4931
29.1449Reserved (RSVD1[8])—Offset 54h 4931
29.1450Reserved (RSVD1[9])—Offset 58h 4932
29.1451Reserved (RSVD1[10])—Offset 5Ch 4932
29.1452GPI Virtual Wire Message Enable (GPI_VWE_scc_0)—Offset 60h 4932
29.1453GPI Virtual Wire Message Enable (GPI_VWE_scc_1)—Offset 64h 4941
29.1454Reserved (RSVD2[0])—Offset 68h 4942
29.1455Reserved (RSVD2[1])—Offset 6Ch 4942
29.1456Reserved (RSVD2[2])—Offset 70h 4942
29.1457Reserved (RSVD2[3])—Offset 74h 4942
29.1458Reserved (RSVD2[4])—Offset 78h 4943
29.1459Reserved (RSVD2[5])—Offset 7Ch 4943

29.1460Pad Configuration Lock (PADCFGLOCK_scc_0)—Offset 80h 4943
29.1461Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_scc_0)—Offset 84h ... 4959
29.1462Pad Configuration Lock (PADCFGLOCK_scc_1)—Offset 88h 4975
29.1463Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_scc_1)—Offset 8Ch ... 4977
29.1464Reserved (RSVD3[0])—Offset 90h 4979
29.1465Reserved (RSVD3[1])—Offset 94h 4979
29.1466Reserved (RSVD3[2])—Offset 98h 4980
29.1467Reserved (RSVD3[3])—Offset 9Ch 4980
29.1468Reserved (RSVD3[4])—Offset A0h 4980
29.1469Reserved (RSVD3[5])—Offset A4h 4980
29.1470Reserved (RSVD3[6])—Offset A8h 4981
29.1471Reserved (RSVD3[7])—Offset ACh 4981
29.1472Host Software Pad Ownership (HOSTSW_OWN_scc_0)—Offset B0h 4981
29.1473Host Software Pad Ownership (HOSTSW_OWN_scc_1)—Offset B4h 4988
29.1474Reserved (RSVD4[0])—Offset B8h 4989
29.1475Reserved (RSVD4[1])—Offset BCh 4989
29.1476Reserved (RSVD4[2])—Offset C0h 4989
29.1477Reserved (RSVD4[3])—Offset C4h 4990
29.1478Reserved (RSVD4[4])—Offset C8h 4990
29.1479Reserved (RSVD4[5])—Offset CCh 4990
29.1480Reserved (RSVD4[6])—Offset D0h 4991
29.1481Reserved (RSVD4[7])—Offset D4h 4991
29.1482Reserved (RSVD4[8])—Offset D8h 4991
29.1483Reserved (RSVD4[9])—Offset DCh 4991
29.1484Reserved (RSVD4[10])—Offset E0h 4992
29.1485Reserved (RSVD4[11])—Offset E4h 4992
29.1486Reserved (RSVD4[12])—Offset E8h 4992
29.1487Reserved (RSVD4[13])—Offset ECh 4992
29.1488Reserved (RSVD4[14])—Offset F0h 4993
29.1489Reserved (RSVD4[15])—Offset F4h 4993
29.1490Reserved (RSVD4[16])—Offset F8h 4993
29.1491Reserved (RSVD4[17])—Offset FCh 4994
29.1492GPI Interrupt Status (GPI_IS_scc_0)—Offset 100h 4994
29.1493GPI Interrupt Status (GPI_IS_scc_1)—Offset 104h 5002
29.1494Reserved (RSVD5[0])—Offset 108h 5003
29.1495Reserved (RSVD5[1])—Offset 10Ch 5003
29.1496GPI Interrupt Enable (GPI_IE_scc_0)—Offset 110h 5004
29.1497GPI Interrupt Enable (GPI_IE_scc_1)—Offset 114h 5009
29.1498Reserved (RSVD6[0])—Offset 118h 5010
29.1499Reserved (RSVD6[1])—Offset 11Ch 5010
29.1500Reserved (RSVD6[2])—Offset 120h 5011
29.1501Reserved (RSVD6[3])—Offset 124h 5011
29.1502Reserved (RSVD6[4])—Offset 128h 5011
29.1503Reserved (RSVD6[5])—Offset 12Ch 5011
29.1504GPI General Purpose Events Status (GPI_GPE_STS_scc_0)—Offset 130h 5012
29.1505GPI General Purpose Events Status (GPI_GPE_STS_scc_1)—Offset 134h 5020
29.1506Reserved (RSVD7[0])—Offset 138h 5021
29.1507Reserved (RSVD7[1])—Offset 13Ch 5022
29.1508Reserved (RSVD7[2])—Offset 140h 5022
29.1509Reserved (RSVD7[3])—Offset 144h 5022
29.1510Reserved (RSVD7[4])—Offset 148h 5022
29.1511Reserved (RSVD7[5])—Offset 14Ch 5023
29.1512GPI General Purpose Events Enable (GPI_GPE_EN_scc_0)—Offset 150h 5023

29.1513GPI General Purpose Events Enable (GPI_GPE_EN_scc_1)—Offset 154h 5030
29.1514Reserved (RSVD8[0])—Offset 158h 5030
29.1515Reserved (RSVD8[1])—Offset 15Ch 5031
29.1516Reserved (RSVD8[2])—Offset 160h 5031
29.1517Reserved (RSVD8[3])—Offset 164h 5031
29.1518Reserved (RSVD8[4])—Offset 168h 5031
29.1519Reserved (RSVD8[5])—Offset 16Ch 5032
29.1520SMI Status (GPI_SMI_STS_scc_0)—Offset 170h.....5032
29.1521SMI Status (GPI_SMI_STS_scc_1)—Offset 174h.....5043
29.1522Reserved (RSVD9[0])—Offset 178h 5044
29.1523Reserved (RSVD9[1])—Offset 17Ch 5044
29.1524Reserved (RSVD9[2])—Offset 180h 5045
29.1525Reserved (RSVD9[3])—Offset 184h 5045
29.1526Reserved (RSVD9[4])—Offset 188h 5045
29.1527Reserved (RSVD9[5])—Offset 18Ch 5046
29.1528SMI Enable (GPI_SMI_EN_scc_0)—Offset 190h 5046
29.1529SMI Enable (GPI_SMI_EN_scc_1)—Offset 194h 5054
29.1530Reserved (RSVD10[0])—Offset 198h 5055
29.1531Reserved (RSVD10[1])—Offset 19Ch 5055
29.1532Reserved (RSVD10[2])—Offset 1A0h 5056
29.1533Reserved (RSVD10[3])—Offset 1A4h 5056
29.1534Reserved (RSVD10[4])—Offset 1A8h 5056
29.1535Reserved (RSVD10[5])—Offset 1ACh 5056
29.1536NMI Status (GPI_NMI_STS_scc_0)—Offset 1B0h 5057
29.1537NMI Status (GPI_NMI_STS_scc_1)—Offset 1B4h 5065
29.1538Reserved (RSVD11[0])—Offset 1B8h 5066
29.1539Reserved (RSVD11[1])—Offset 1BCh 5066
29.1540Reserved (RSVD11[2])—Offset 1C0h 5067
29.1541Reserved (RSVD11[3])—Offset 1C4h 5067
29.1542Reserved (RSVD11[4])—Offset 1C8h 5067
29.1543Reserved (RSVD11[5])—Offset 1CCh 5067
29.1544NMI Enable (GPI_NMI_EN_scc_0)—Offset 1D0h 5068
29.1545NMI Enable (GPI_NMI_EN_scc_1)—Offset 1D4h 5076
29.1546Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_SMBUS_3P3)—Offset 300h 5077
29.1547Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_SMBUS_3P3)—Offset 304h 5077
29.1548Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_SMBUS_3P3)—Offset 308h 5078
29.1549Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_SMBUS_3P3)—Offset 30Ch 5078
29.1550Family Configuration Register (FAM_CFG_SMBUS_3P3)—Offset 310h 5079
29.1551Family Reserved Register DW5 (FAM_RSVD_DW5SMBUS_3P3)—Offset 314h 5080
29.1552Family Reserved Register DW6 (FAM_RSVD_DW6SMBUS_3P3)—Offset 318h 5080
29.1553Family Reserved Register DW7 (FAM_RSVD_DW7SMBUS_3P3)—Offset 31Ch 5081
29.1554Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_SDCARD_3P3)—Offset 320h 5081
29.1555Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_SDCARD_3P3)—Offset 324h 5081
29.1556Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_SDCARD_3P3)—Offset 328h 5082
29.1557Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_SDCARD_3P3)—Offset 32Ch 5082
29.1558Family Configuration Register (FAM_CFG_SDCARD_3P3)—Offset 330h 5083
29.1559Family Reserved Register DW5 (FAM_RSVD_DW5SDCARD_3P3)—Offset 334h.....5084
29.1560Family Reserved Register DW6 (FAM_RSVD_DW6SDCARD_3P3)—Offset 338h.....5085
29.1561Family Reserved Register DW7 (FAM_RSVD_DW7SDCARD_3P3)—Offset 33Ch 5085
29.1562Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_SD_CLK_1P8)—Offset 340h..... 5085

29.1563Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_SD_CLK_1P8)—Offset 344h 5086
29.1564Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_SD_CLK_1P8)—Offset 348h 5086
29.1565Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_SD_CLK_1P8)—Offset 34Ch 5087
29.1566Family Configuration Register (FAM_CFG_SD_CLK_1P8)—Offset 350h 5087
29.1567Family Reserved Register DW5 (FAM_RSVD_DW5SD_CLK_1P8)—Offset 354h..... 5089
29.1568Family Reserved Register DW6 (FAM_RSVD_DW6SD_CLK_1P8)—Offset 358h..... 5089
29.1569Family Reserved Register DW7 (FAM_RSVD_DW7SD_CLK_1P8)—Offset 35Ch..... 5089
29.1570Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_CNV_1P8)—Offset 360h ... 5090
29.1571Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_CNV_1P8)—Offset 364h ... 5090
29.1572Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_CNV_1P8)—Offset 368h ... 5091
29.1573Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_CNV_1P8)—Offset 36Ch ... 5091
29.1574Family Configuration Register (FAM_CFG_CNV_1P8)—Offset 370h..... 5091
29.1575Family Reserved Register DW5 (FAM_RSVD_DW5CNV_1P8)—Offset 374h 5093
29.1576Family Reserved Register DW6 (FAM_RSVD_DW6CNV_1P8)—Offset 378h 5093
29.1577Family Reserved Register DW7 (FAM_RSVD_DW7CNV_1P8)—Offset 37Ch 5094
29.1578Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_EMMC_1P8)—Offset 380h . 5094
29.1579Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_EMMC_1P8)—Offset 384h . 5094
29.1580Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_EMMC_1P8)—Offset 388h . 5095
29.1581Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_EMMC_1P8)—Offset 38Ch . 5095
29.1582Family Configuration Register (FAM_CFG_EMMC_1P8)—Offset 390h 5096
29.1583Family Reserved Register DW5 (FAM_RSVD_DW5EMMC_1P8)—Offset 394h..... 5097
29.1584Family Reserved Register DW6 (FAM_RSVD_DW6EMMC_1P8)—Offset 398h..... 5098
29.1585Family Reserved Register DW7 (FAM_RSVD_DW7EMMC_1P8)—Offset 39Ch 5098
29.1586RCOMP Control Register (RCOMP_CTL_SMBUS_3P3)—Offset 480h 5098
29.1587RCOMP Offset Register (RCOMP_OFFSET_SMBUS_3P3)—Offset 484h..... 5099
29.1588RCOMP Override Registers DW0 (RCOMP_OVR_DW0_SMBUS_3P3)—Offset 488h . 5100
29.1589RCOMP Override Registers DW1 (RCOMP_OVR_DW1_SMBUS_3P3)—Offset 48Ch . 5101
29.1590RCOMP Value Register DW0 (RCOMP_VALUE_DW0_SMBUS_3P3)—Offset 490h 5101
29.1591RCOMP Value Register DW1 (RCOMP_VALUE_DW1_SMBUS_3P3)—Offset 494h 5102
29.1592RCOMP Control Register DW1 (RCOMP_CTL_DW1_SMBUS_3P3)—Offset 498h 5102
29.1593RCOMP Reserved Register (RCOMP_RSVD_SMBUS_3P3)—Offset 49Ch..... 5103
29.1594RCOMP Control Register (RCOMP_CTL_SDCARD_3P3)—Offset 4A0h 5104
29.1595RCOMP Offset Register (RCOMP_OFFSET_SDCARD_3P3)—Offset 4A4h..... 5105
29.1596RCOMP Override Registers DW0 (RCOMP_OVR_DW0_SDCARD_3P3)—Offset 4A8h 5105
29.1597RCOMP Override Registers DW1 (RCOMP_OVR_DW1_SDCARD_3P3)—Offset 4ACh 5106
29.1598RCOMP Value Register DW0 (RCOMP_VALUE_DW0_SDCARD_3P3)—Offset 4B0h .. 5107
29.1599RCOMP Value Register DW1 (RCOMP_VALUE_DW1_SDCARD_3P3)—Offset 4B4h .. 5107
29.1600RCOMP Control Register DW1 (RCOMP_CTL_DW1_SDCARD_3P3)—Offset 4B8h ... 5107
29.1601RCOMP Reserved Register (RCOMP_RSVD_SDCARD_3P3)—Offset 4BCh..... 5109
29.1602RCOMP Control Register (RCOMP_CTL_SD_CLK_1P8)—Offset 4C0h 5109
29.1603RCOMP Offset Register (RCOMP_OFFSET_SD_CLK_1P8)—Offset 4C4h 5110
29.1604RCOMP Override Registers DW0 (RCOMP_OVR_DW0_SD_CLK_1P8)—Offset 4C8h 5111
29.1605RCOMP Override Registers DW1 (RCOMP_OVR_DW1_SD_CLK_1P8)—Offset 4CCh 5111
29.1606RCOMP Value Register DW0 (RCOMP_VALUE_DW0_SD_CLK_1P8)—Offset 4D0h .. 5112
29.1607RCOMP Value Register DW1 (RCOMP_VALUE_DW1_SD_CLK_1P8)—Offset 4D4h .. 5112
29.1608RCOMP Control Register DW1 (RCOMP_CTL_DW1_SD_CLK_1P8)—Offset 4D8h 5113
29.1609RCOMP Reserved Register (RCOMP_RSVD_SD_CLK_1P8)—Offset 4DCh 5114
29.1610RCOMP Control Register (RCOMP_CTL_EMMC_1P8)—Offset 4E0h 5114
29.1611RCOMP Offset Register (RCOMP_OFFSET_EMMC_1P8)—Offset 4E4h 5115
29.1612RCOMP Override Registers DW0 (RCOMP_OVR_DW0_EMMC_1P8)—Offset 4E8h ... 5116
29.1613RCOMP Override Registers DW1 (RCOMP_OVR_DW1_EMMC_1P8)—Offset 4ECh ... 5117
29.1614RCOMP Value Register DW0 (RCOMP_VALUE_DW0_EMMC_1P8)—Offset 4F0h..... 5117

29.1615RCOMP Value Register DW1 (RCOMP_VALUE_DW1_EMMC_1P8)—Offset 4F4h 5118
29.1616RCOMP Control Register DW1 (RCOMP_CTL_DW1_EMMC_1P8)—Offset 4F8h 5118
29.1617RCOMP Reserved Register (RCOMP_RSVD_EMMC_1P8)—Offset 4FCh 5119
29.1618Pad Configuration DW0 (PAD_CFG_DW0_GPIO_176)—Offset 600h 5120
29.1619Pad Configuration DW1 (PAD_CFG_DW1_GPIO_176)—Offset 604h 5122
29.1620Pad Configuration DW2 (PAD_CFG_DW2_GPIO_176)—Offset 608h 5124
29.1621Pad Configuration DW3 (PAD_CFG_DW3_GPIO_176)—Offset 60Ch 5125
29.1622Pad Configuration DW0 (PAD_CFG_DW0_GPIO_177)—Offset 610h 5125
29.1623Pad Configuration DW1 (PAD_CFG_DW1_GPIO_177)—Offset 614h 5127
29.1624Pad Configuration DW2 (PAD_CFG_DW2_GPIO_177)—Offset 618h 5129
29.1625Pad Configuration DW3 (PAD_CFG_DW3_GPIO_177)—Offset 61Ch 5130
29.1626Pad Configuration DW0 (PAD_CFG_DW0_GPIO_178)—Offset 620h 5130
29.1627Pad Configuration DW1 (PAD_CFG_DW1_GPIO_178)—Offset 624h 5132
29.1628Pad Configuration DW2 (PAD_CFG_DW2_GPIO_178)—Offset 628h 5134
29.1629Pad Configuration DW3 (PAD_CFG_DW3_GPIO_178)—Offset 62Ch 5135
29.1630Pad Configuration DW0 (PAD_CFG_DW0_GPIO_187)—Offset 630h 5135
29.1631Pad Configuration DW1 (PAD_CFG_DW1_GPIO_187)—Offset 634h 5137
29.1632Pad Configuration DW2 (PAD_CFG_DW2_GPIO_187)—Offset 638h 5139
29.1633Pad Configuration DW3 (PAD_CFG_DW3_GPIO_187)—Offset 63Ch 5140
29.1634Pad Configuration DW0 (PAD_CFG_DW0_GPIO_179)—Offset 640h 5140
29.1635Pad Configuration DW1 (PAD_CFG_DW1_GPIO_179)—Offset 644h 5142
29.1636Pad Configuration DW2 (PAD_CFG_DW2_GPIO_179)—Offset 648h 5144
29.1637Pad Configuration DW3 (PAD_CFG_DW3_GPIO_179)—Offset 64Ch 5145
29.1638Pad Configuration DW0 (PAD_CFG_DW0_GPIO_180)—Offset 650h 5145
29.1639Pad Configuration DW1 (PAD_CFG_DW1_GPIO_180)—Offset 654h 5147
29.1640Pad Configuration DW2 (PAD_CFG_DW2_GPIO_180)—Offset 658h 5149
29.1641Pad Configuration DW3 (PAD_CFG_DW3_GPIO_180)—Offset 65Ch 5150
29.1642Pad Configuration DW0 (PAD_CFG_DW0_GPIO_181)—Offset 660h 5150
29.1643Pad Configuration DW1 (PAD_CFG_DW1_GPIO_181)—Offset 664h 5152
29.1644Pad Configuration DW2 (PAD_CFG_DW2_GPIO_181)—Offset 668h 5154
29.1645Pad Configuration DW3 (PAD_CFG_DW3_GPIO_181)—Offset 66Ch 5155
29.1646Pad Configuration DW0 (PAD_CFG_DW0_GPIO_182)—Offset 670h 5155
29.1647Pad Configuration DW1 (PAD_CFG_DW1_GPIO_182)—Offset 674h 5157
29.1648Pad Configuration DW2 (PAD_CFG_DW2_GPIO_182)—Offset 678h 5159
29.1649Pad Configuration DW3 (PAD_CFG_DW3_GPIO_182)—Offset 67Ch 5160
29.1650Pad Configuration DW0 (PAD_CFG_DW0_GPIO_183)—Offset 680h 5160
29.1651Pad Configuration DW1 (PAD_CFG_DW1_GPIO_183)—Offset 684h 5162
29.1652Pad Configuration DW2 (PAD_CFG_DW2_GPIO_183)—Offset 688h 5164
29.1653Pad Configuration DW3 (PAD_CFG_DW3_GPIO_183)—Offset 68Ch 5165
29.1654Pad Configuration DW0 (PAD_CFG_DW0_GPIO_184)—Offset 690h 5165
29.1655Pad Configuration DW1 (PAD_CFG_DW1_GPIO_184)—Offset 694h 5167
29.1656Pad Configuration DW2 (PAD_CFG_DW2_GPIO_184)—Offset 698h 5169
29.1657Pad Configuration DW3 (PAD_CFG_DW3_GPIO_184)—Offset 69Ch 5170
29.1658Pad Configuration DW0 (PAD_CFG_DW0_GPIO_185)—Offset 6A0h 5170
29.1659Pad Configuration DW1 (PAD_CFG_DW1_GPIO_185)—Offset 6A4h 5172
29.1660Pad Configuration DW2 (PAD_CFG_DW2_GPIO_185)—Offset 6A8h 5174
29.1661Pad Configuration DW3 (PAD_CFG_DW3_GPIO_185)—Offset 6ACh 5175
29.1662Pad Configuration DW0 (PAD_CFG_DW0_GPIO_186)—Offset 6B0h 5175
29.1663Pad Configuration DW1 (PAD_CFG_DW1_GPIO_186)—Offset 6B4h 5177
29.1664Pad Configuration DW2 (PAD_CFG_DW2_GPIO_186)—Offset 6B8h 5179
29.1665Pad Configuration DW3 (PAD_CFG_DW3_GPIO_186)—Offset 6BCh 5180
29.1666Pad Configuration DW0 (PAD_CFG_DW0_GPIO_188)—Offset 6C0h 5180
29.1667Pad Configuration DW1 (PAD_CFG_DW1_GPIO_188)—Offset 6C4h 5182
29.1668Pad Configuration DW2 (PAD_CFG_DW2_GPIO_188)—Offset 6C8h 5184
29.1669Pad Configuration DW3 (PAD_CFG_DW3_GPIO_188)—Offset 6CCh 5185

29.1670Pad Configuration DW0 (PAD_CFG_DW0_GPIO_210)—Offset 6D0h 5185
29.1671Pad Configuration DW1 (PAD_CFG_DW1_GPIO_210)—Offset 6D4h 5187
29.1672Pad Configuration DW2 (PAD_CFG_DW2_GPIO_210)—Offset 6D8h 5189
29.1673Pad Configuration DW3 (PAD_CFG_DW3_GPIO_210)—Offset 6DCh 5190
29.1674Pad Configuration DW0 (PAD_CFG_DW0_GPIO_189)—Offset 6E0h..... 5190
29.1675Pad Configuration DW1 (PAD_CFG_DW1_GPIO_189)—Offset 6E4h..... 5192
29.1676Pad Configuration DW2 (PAD_CFG_DW2_GPIO_189)—Offset 6E8h..... 5194
29.1677Pad Configuration DW3 (PAD_CFG_DW3_GPIO_189)—Offset 6ECh..... 5195
29.1678Pad Configuration DW0 (PAD_CFG_DW0_GPIO_190)—Offset 6F0h..... 5195
29.1679Pad Configuration DW1 (PAD_CFG_DW1_GPIO_190)—Offset 6F4h..... 5197
29.1680Pad Configuration DW2 (PAD_CFG_DW2_GPIO_190)—Offset 6F8h..... 5199
29.1681Pad Configuration DW3 (PAD_CFG_DW3_GPIO_190)—Offset 6FCh..... 5200
29.1682Pad Configuration DW0 (PAD_CFG_DW0_GPIO_191)—Offset 700h..... 5200
29.1683Pad Configuration DW1 (PAD_CFG_DW1_GPIO_191)—Offset 704h..... 5202
29.1684Pad Configuration DW2 (PAD_CFG_DW2_GPIO_191)—Offset 708h..... 5204
29.1685Pad Configuration DW3 (PAD_CFG_DW3_GPIO_191)—Offset 70Ch..... 5205
29.1686Pad Configuration DW0 (PAD_CFG_DW0_GPIO_192)—Offset 710h..... 5205
29.1687Pad Configuration DW1 (PAD_CFG_DW1_GPIO_192)—Offset 714h..... 5207
29.1688Pad Configuration DW2 (PAD_CFG_DW2_GPIO_192)—Offset 718h..... 5209
29.1689Pad Configuration DW3 (PAD_CFG_DW3_GPIO_192)—Offset 71Ch..... 5210
29.1690Pad Configuration DW0 (PAD_CFG_DW0_GPIO_193)—Offset 720h..... 5210
29.1691Pad Configuration DW1 (PAD_CFG_DW1_GPIO_193)—Offset 724h..... 5212
29.1692Pad Configuration DW2 (PAD_CFG_DW2_GPIO_193)—Offset 728h..... 5214
29.1693Pad Configuration DW3 (PAD_CFG_DW3_GPIO_193)—Offset 72Ch..... 5215
29.1694Pad Configuration DW0 (PAD_CFG_DW0_GPIO_194)—Offset 730h..... 5215
29.1695Pad Configuration DW1 (PAD_CFG_DW1_GPIO_194)—Offset 734h..... 5217
29.1696Pad Configuration DW2 (PAD_CFG_DW2_GPIO_194)—Offset 738h..... 5219
29.1697Pad Configuration DW3 (PAD_CFG_DW3_GPIO_194)—Offset 73Ch..... 5220
29.1698Pad Configuration DW0 (PAD_CFG_DW0_GPIO_195)—Offset 740h..... 5220
29.1699Pad Configuration DW1 (PAD_CFG_DW1_GPIO_195)—Offset 744h..... 5222
29.1700Pad Configuration DW2 (PAD_CFG_DW2_GPIO_195)—Offset 748h..... 5224
29.1701Pad Configuration DW3 (PAD_CFG_DW3_GPIO_195)—Offset 74Ch..... 5225
29.1702Pad Configuration DW0 (PAD_CFG_DW0_GPIO_196)—Offset 750h..... 5225
29.1703Pad Configuration DW1 (PAD_CFG_DW1_GPIO_196)—Offset 754h..... 5227
29.1704Pad Configuration DW2 (PAD_CFG_DW2_GPIO_196)—Offset 758h..... 5229
29.1705Pad Configuration DW3 (PAD_CFG_DW3_GPIO_196)—Offset 75Ch..... 5230
29.1706Pad Configuration DW0 (PAD_CFG_DW0_GPIO_197)—Offset 760h..... 5230
29.1707Pad Configuration DW1 (PAD_CFG_DW1_GPIO_197)—Offset 764h..... 5232
29.1708Pad Configuration DW2 (PAD_CFG_DW2_GPIO_197)—Offset 768h..... 5234
29.1709Pad Configuration DW3 (PAD_CFG_DW3_GPIO_197)—Offset 76Ch..... 5235
29.1710Pad Configuration DW0 (PAD_CFG_DW0_GPIO_198)—Offset 770h..... 5235
29.1711Pad Configuration DW1 (PAD_CFG_DW1_GPIO_198)—Offset 774h..... 5237
29.1712Pad Configuration DW2 (PAD_CFG_DW2_GPIO_198)—Offset 778h..... 5239
29.1713Pad Configuration DW3 (PAD_CFG_DW3_GPIO_198)—Offset 77Ch..... 5240
29.1714Pad Configuration DW0 (PAD_CFG_DW0_GPIO_199)—Offset 780h..... 5240
29.1715Pad Configuration DW1 (PAD_CFG_DW1_GPIO_199)—Offset 784h..... 5242
29.1716Pad Configuration DW2 (PAD_CFG_DW2_GPIO_199)—Offset 788h..... 5244
29.1717Pad Configuration DW3 (PAD_CFG_DW3_GPIO_199)—Offset 78Ch..... 5245
29.1718Pad Configuration DW0 (PAD_CFG_DW0_GPIO_200)—Offset 790h..... 5245
29.1719Pad Configuration DW1 (PAD_CFG_DW1_GPIO_200)—Offset 794h..... 5247
29.1720Pad Configuration DW2 (PAD_CFG_DW2_GPIO_200)—Offset 798h..... 5249
29.1721Pad Configuration DW3 (PAD_CFG_DW3_GPIO_200)—Offset 79Ch..... 5250
29.1722Pad Configuration DW0 (PAD_CFG_DW0_GPIO_201)—Offset 7A0h..... 5250
29.1723Pad Configuration DW1 (PAD_CFG_DW1_GPIO_201)—Offset 7A4h..... 5252
29.1724Pad Configuration DW2 (PAD_CFG_DW2_GPIO_201)—Offset 7A8h..... 5254

29.1725Pad Configuration DW3 (PAD_CFG_DW3_GPIO_201)—Offset 7ACh 5255
29.1726Pad Configuration DW0 (PAD_CFG_DW0_GPIO_202)—Offset 7B0h 5255
29.1727Pad Configuration DW1 (PAD_CFG_DW1_GPIO_202)—Offset 7B4h 5257
29.1728Pad Configuration DW2 (PAD_CFG_DW2_GPIO_202)—Offset 7B8h 5259
29.1729Pad Configuration DW3 (PAD_CFG_DW3_GPIO_202)—Offset 7BCh 5260
29.1730Pad Configuration DW0 (PAD_CFG_DW0_GPIO_203)—Offset 7C0h 5260
29.1731Pad Configuration DW1 (PAD_CFG_DW1_GPIO_203)—Offset 7C4h 5262
29.1732Pad Configuration DW2 (PAD_CFG_DW2_GPIO_203)—Offset 7C8h 5264
29.1733Pad Configuration DW3 (PAD_CFG_DW3_GPIO_203)—Offset 7CCh 5265
29.1734Pad Configuration DW0 (PAD_CFG_DW0_GPIO_204)—Offset 7D0h 5265
29.1735Pad Configuration DW1 (PAD_CFG_DW1_GPIO_204)—Offset 7D4h 5267
29.1736Pad Configuration DW2 (PAD_CFG_DW2_GPIO_204)—Offset 7D8h 5269
29.1737Pad Configuration DW3 (PAD_CFG_DW3_GPIO_204)—Offset 7DCh 5270
29.1738Pad Configuration DW0 (PAD_CFG_DW0_GPIO_205)—Offset 7E0h 5270
29.1739Pad Configuration DW1 (PAD_CFG_DW1_GPIO_205)—Offset 7E4h 5272
29.1740Pad Configuration DW2 (PAD_CFG_DW2_GPIO_205)—Offset 7E8h 5274
29.1741Pad Configuration DW3 (PAD_CFG_DW3_GPIO_205)—Offset 7ECh 5275
29.1742Pad Configuration DW0 (PAD_CFG_DW0_GPIO_206)—Offset 7F0h 5275
29.1743Pad Configuration DW1 (PAD_CFG_DW1_GPIO_206)—Offset 7F4h 5277
29.1744Pad Configuration DW2 (PAD_CFG_DW2_GPIO_206)—Offset 7F8h 5279
29.1745Pad Configuration DW3 (PAD_CFG_DW3_GPIO_206)—Offset 7FCh 5280
29.1746Pad Configuration DW0 (PAD_CFG_DW0_GPIO_207)—Offset 800h 5280
29.1747Pad Configuration DW1 (PAD_CFG_DW1_GPIO_207)—Offset 804h 5282
29.1748Pad Configuration DW2 (PAD_CFG_DW2_GPIO_207)—Offset 808h 5284
29.1749Pad Configuration DW3 (PAD_CFG_DW3_GPIO_207)—Offset 80Ch 5285
29.1750Pad Configuration DW0 (PAD_CFG_DW0_GPIO_208)—Offset 810h 5285
29.1751Pad Configuration DW1 (PAD_CFG_DW1_GPIO_208)—Offset 814h 5287
29.1752Pad Configuration DW2 (PAD_CFG_DW2_GPIO_208)—Offset 818h 5289
29.1753Pad Configuration DW3 (PAD_CFG_DW3_GPIO_208)—Offset 81Ch 5290
29.1754Pad Configuration DW0 (PAD_CFG_DW0_GPIO_209)—Offset 820h 5290
29.1755Pad Configuration DW1 (PAD_CFG_DW1_GPIO_209)—Offset 824h 5292
29.1756Pad Configuration DW2 (PAD_CFG_DW2_GPIO_209)—Offset 828h 5294
29.1757Pad Configuration DW3 (PAD_CFG_DW3_GPIO_209)—Offset 82Ch 5295

30 MSR Registers 5296

30.1 IA32_MC5_CTL (P_CR_MC_CTL)—Offset 4620h 5296
30.2 IA32_MC5_STATUS (P_CR_MC_STATUS)—Offset 4628h 5296
30.3 IA32_MC5_CTL2 (P_CR_MC_CTL2)—Offset 4630h 5297
30.4 VR_MISC_CONFIG (P_CR_VR_MISC_CONFIG)—Offset 4670h 5298
30.5 Thread T-state Request (P_CR_THREAD_T_REQ)—Offset 46B8h 5299
30.6 IA32_MC5_ADDR (P_CR_MC_ADDR)—Offset 46E0h 5299
30.7 IA32_MC5_MISC (P_CR_MC_MISC)—Offset 46E8h.....5299
30.8 PLATFORM_ID (P_CR_PLATFORM_ID)—Offset 4710h.....5300
30.9 PLATFORM_INFO (P_CR_PLATFORM_INFO)—Offset 4718h 5300
30.10 DDR_RAPL_LIMIT (P_CR_DDR_RAPL_LIMIT)—Offset 4760h 5302
30.11 FREQ_LIMIT_RATIOS (P_CR_FREQ_LIMIT_RATIOS)—Offset 4768h 5302
30.12 FREQ_LIMIT_CORES (P_CR_FREQ_LIMIT_CORES)—Offset 4770h.....5303
30.13 Package RAPL Power Limit (P_CR_PACKAGE_RAPL_LIMIT)—Offset 4780h 5304
30.14 PL3_CONTROL_MSR (P_CR_PL3_CONTROL)—Offset 4790h 5306
30.15 VR_CURRENT_CONFIG (P_CR_VR_CURRENT_CONFIG)—Offset 47C0h 5306
30.16 Core Thermal and Power Status (P_CR_THERM_STATUS)—Offset 4828h.....5307
30.17 Thermal Interrupt Control (P_CR_THERM_INTERRUPT)—Offset 482Ch 5309
30.18 C1E Idle Residency (P_CR_PC2_RCNTR)—Offset 4870h 5310
30.19 PC3 Idle Residency Counter (P_CR_PC3_RCNTR)—Offset 4878h 5310
30.20 PCS_Idle_Residency (P_CR_PC6_RCNTR)—Offset 4880h.....5311

30.21 PCS S0i3 Residency (P_CR_PC7_RCNTR)—Offset 4888h 5311
30.22 SOC S0i3 Residency (P_CR_PC10_RCNTR)—Offset 4890h 5312
30.23 LLC_FLUSHED_RCNTR (P_CR_L2_FLUSH_RCNTR)—Offset 4898h 5312
30.24 PC9 Idle Residency Counter (P_CR_PC9_RCNTR)—Offset 48A0h 5313
30.25 C_STATE_LATENCY_CONTROL_0 (P_CR_C_STATE_LATENCY_CONTROL_0)—Offset 48A8h 5313
30.26 C_STATE_LATENCY_CONTROL_1 (P_CR_C_STATE_LATENCY_CONTROL_1)—Offset 48ACh 5314
30.27 C_STATE_LATENCY_CONTROL_2 (P_CR_C_STATE_LATENCY_CONTROL_2)—Offset 48B0h 5315
30.28 C_STATE_LATENCY_CONTROL_3 (P_CR_C_STATE_LATENCY_CONTROL_3)—Offset 48B4h 5316
30.29 C_STATE_LATENCY_CONTROL_4 (P_CR_C_STATE_LATENCY_CONTROL_4)—Offset 48B8h 5316
30.30 C_STATE_LATENCY_CONTROL_5 (P_CR_C_STATE_LATENCY_CONTROL_5)—Offset 48BCh 5317
30.31 Package Thermal and Power Status (P_CR_PACKAGE_THERM_STATUS)—Offset 49F0h... 5318
30.32 Thermal Interrupt Control (P_CR_PACKAGE_THERM_INTERRUPT)—Offset 49F4h 5319
30.33 IA32_MC6_STATUS (A_CR_MCI_STATUS)—Offset 4F80h 5320
30.34 MCA Control Register (A_CR_MCI_CTL)—Offset 4F88h 5321
30.35 IA32_MC6_CTL2 (A_CR_MCI_CTL2)—Offset 4F90h 5323
30.36 IA32_MC6_ADDR (A_CR_MCI_ADDR)—Offset 4F98h 5323
30.37 IA32_MC6_MISC (A_CR_MCI_MISC)—Offset 4FC0h 5324
30.38 IA32_MC4_CTL (B_CR_MCI_CTL)—Offset 4B20h 5324
30.39 IA32_MC4_STATUS (B_CR_MCI_STATUS)—Offset 4B28h 5325
30.40 IA32_MC4_ADDR (B_CR_MCI_ADDR)—Offset 4B30h 5326
30.41 IA32_MC4_MISC (B_CR_MCI_MISC)—Offset 4B38h 5326
30.42 IA32_MC4_CTL2 (B_CR_MCI_CTL2)—Offset 4B40h 5327
30.43 B-Unit Slice and Channel Hash Function (B_CR_SLICE_CHANNEL_HASH)—Offset 4C58h. 5327
30.44 SMM_BLOCKED (T_CR_SMM_BLOCKED)—Offset 5100h 5329
30.45 SMM_DELAYED (T_CR_SMM_DELAYED)—Offset 5108h 5330

31 MODPHY Registers 5331

31.1 CLK_SRC_SELECT Register (CLK_SRC_SELECT)—Offset 0h 5331
31.2 TXESC_CLK_DIV1 Register (TXESC_CLK_DIV1)—Offset 4h 5331
31.3 TXESC_CLK_DIV2 Register (TXESC_CLK_DIV2)—Offset 8h 5331
31.4 RCOMP_CLK_DIV1 Register (RCOMP_CLK_DIV1)—Offset Ch 5332
31.5 RCOMP_CLK_DIV2 Register (RCOMP_CLK_DIV2)—Offset 10h 5332
31.6 TOP_CFG_REG Register (TOP_CFG_REG)—Offset 14h 5333
31.7 REF_CTL_REG Register (REF_CTL_REG)—Offset 18h 5333
31.8 DLY_ADJ_REG Register (DLY_ADJ_REG)—Offset 1Ch 5334
31.9 HS_RCOMP_READOUT Register (HS_RCOMP_READOUT)—Offset 20h 5334
31.10 LP_P_RCOMP_READOUT Register (LP_P_RCOMP_READOUT)—Offset 24h 5335
31.11 LP_N_RCOMP_READOUT Register (LP_N_RCOMP_READOUT)—Offset 28h 5335
31.12 RCOMP_LP_SLEW_CFG Register (RCOMP_LP_SLEW_CFG)—Offset 2Ch 5336
31.13 CLK_MONITOR_PORT_CONTROL Register (CLK_MONITOR_PORT_CONTROL)—Offset 30h 5337
31.14 HS_RCOMP_CFG Register (HS_RCOMP_CFG)—Offset 34h 5337
31.15 LP_RCOMP_CFG Register (LP_RCOMP_CFG)—Offset 38h 5338
31.16 RCOMP_STAT Register (RCOMP_STAT)—Offset 3Ch 5339
31.17 RCCRCOMP Register (RCCRCOMP)—Offset 40h 5339
31.18 RCOMPCTL Register (RCOMPCTL)—Offset 44h 5340
31.19 PWRMNG_OVR_CTL Register (PWRMNG_OVR_CTL)—Offset 48h 5341
31.20 PWRMNG_STATUS Register (PWRMNG_STATUS)—Offset 4Ch 5342

31.21 GLOBAL_GLK_RESERVED3 Register (GLOBAL_GLK_RESERVED3)—Offset 50h 5343
31.22 GLOBAL_RESERVED1 Register (GLOBAL_RESERVED1)—Offset 54h 5343
31.23 TX_DFX_TOP_CFG_REG Register (TX_DFX_TOP_CFG_REG)—Offset 58h 5344
31.24 TX_DFX_TOP_RESERVED1 Register (TX_DFX_TOP_RESERVED1)—Offset 5Ch 5344
31.25 TX_DFX_TOP_RESERVED2 Register (TX_DFX_TOP_RESERVED2)—Offset 60h 5345
31.26 TX_DFX_TOP_RESERVED3 Register (TX_DFX_TOP_RESERVED3)—Offset 64h 5345
31.27 TX_DFX_TOP_RESERVED4 Register (TX_DFX_TOP_RESERVED4)—Offset 68h 5345
31.28 TX_DFX_TOP_RESERVED5 Register (TX_DFX_TOP_RESERVED5)—Offset 6Ch 5346
31.29 TX_DFX_TOP_RESERVED6 Register (TX_DFX_TOP_RESERVED6)—Offset 70h 5346
31.30 TX_DFX_TOP_RESERVED7 Register (TX_DFX_TOP_RESERVED7)—Offset 74h 5346
31.31 TX_DFX_TOP_RESERVED8 Register (TX_DFX_TOP_RESERVED8)—Offset 78h 5347
31.32 TX_DFX_TOP_RESERVED9 Register (TX_DFX_TOP_RESERVED9)—Offset 7Ch 5347
31.33 TX_DFX_PORT0_CTL_REG Register (TX_DFX_PORT0_CTL_REG)—Offset 80h 5347
31.34 TX_DFX_PORT0_CD_COUNTUP_STATUS Register (TX_DFX_PORT0_CD_COUNTUP_STATUS)—Offset 84h 5348
31.35 TX_DFX_PORT0_RCC_CFG_REG Register (TX_DFX_PORT0_RCC_CFG_REG)—Offset 88h . 5349
31.36 TX_DFX_PORT0_ACIO_CFG_REG Register (TX_DFX_PORT0_ACIO_CFG_REG)—Offset 8Ch 5349
31.37 TX_DFX_PORT0_ACIOSS_REG Register (TX_DFX_PORT0_ACIOSS_REG)—Offset 90h 5350
31.38 TX_DFX_PORT0_ACIOERR_REG Register (TX_DFX_PORT0_ACIOERR_REG)—Offset 94h .. 5351
31.39 TX_DFX_PORT0_DELAY_CALIB_REG Register (TX_DFX_PORT0_DELAY_CALIB_REG)—Offset 98h 5352
31.40 TX_DFX_PORT0_RXLPBK_DATA Register (TX_DFX_PORT0_RXLPBK_DATA)—Offset 9Ch .. 5352
31.41 TX_DFX_PORT0_CLANE_CTL_REG Register (TX_DFX_PORT0_CLANE_CTL_REG)—Offset A0h 5352
31.42 TX_DFX_PORT0_RESERVED1 Register (TX_DFX_PORT0_RESERVED1)—Offset A4h ..5353
31.43 TX_DFX_PORT0_RESERVED2 Register (TX_DFX_PORT0_RESERVED2)—Offset A8h ..5353
31.44 TX_DFX_PORT0_RESERVED3 Register (TX_DFX_PORT0_RESERVED3)—Offset ACh ..5353
31.45 TX_DFX_PORT0_DLANE0_CTL_REG Register (TX_DFX_PORT0_DLANE0_CTL_REG)—Offset B0h 5354
31.46 TX_DFX_PORT0_DLANE0_STAT_REG Register (TX_DFX_PORT0_DLANE0_STAT_REG)—Offset B4h 5354
31.47 TX_DFX_PORT0_DLANE1_CTL_REG Register (TX_DFX_PORT0_DLANE1_CTL_REG)—Offset B8h 5355
31.48 TX_DFX_PORT0_DLANE1_STAT_REG Register (TX_DFX_PORT0_DLANE1_STAT_REG)—Offset BCh 5355
31.49 TX_DFX_PORT0_DLANE2_CTL_REG Register (TX_DFX_PORT0_DLANE2_CTL_REG)—Offset C0h 5356
31.50 TX_DFX_PORT0_DLANE2_STAT_REG Register (TX_DFX_PORT0_DLANE2_STAT_REG)—Offset C4h 5356
31.51 TX_DFX_PORT0_DLANE3_CTL_REG Register (TX_DFX_PORT0_DLANE3_CTL_REG)—Offset C8h 5357
31.52 TX_DFX_PORT0_DLANE3_STAT_REG Register (TX_DFX_PORT0_DLANE3_STAT_REG)—Offset CCh 5357
31.53 TX_DFX_PORT0_LANE_RESERVED5 Register (TX_DFX_PORT0_LANE_RESERVED5)—Offset D0h 5358
31.54 TX_DFX_PORT0_LANE_RESERVED6 Register (TX_DFX_PORT0_LANE_RESERVED6)—Offset D4h 5358
31.55 TX_DFX_PORT0_LANE_RESERVED7 Register (TX_DFX_PORT0_LANE_RESERVED7)—Offset D8h 5358
31.56 TX_DFX_PORT0_LANE_RESERVED8 Register (TX_DFX_PORT0_LANE_RESERVED8)—Offset DCh 5359

31.57 TX_DFE_PORT0_DLN_CNT_LPX Register (TX_DFE_PORT0_DLN_CNT_LPX)—Offset E0h .. 5359
31.58 TX_DFE_PORT0_DLN_CNT_HS_PREP Register (TX_DFE_PORT0_DLN_CNT_HS_PREP)—Offset E4h 5359
31.59 TX_DFE_PORT0_DLN_CNT_HS_ZERO Register (TX_DFE_PORT0_DLN_CNT_HS_ZERO)—Offset E8h..... 5360
31.60 TX_DFE_PORT0_DLN_CNT_HS_TRAIL Register (TX_DFE_PORT0_DLN_CNT_HS_TRAIL)—Offset ECh 5360
31.61 TX_DFE_PORT0_DLN_CNT_HS_EXIT Register (TX_DFE_PORT0_DLN_CNT_HS_EXIT)—Offset F0h 5361
31.62 TX_DFE_PORT0_DLN_CNT_RX Register (TX_DFE_PORT0_DLN_CNT_RX)—Offset F4h..... 5361
31.63 TX_DFE_PORT0_DLN_SYNC_CNT Register (TX_DFE_PORT0_DLN_SYNC_CNT)—Offset F8h..... 5361
31.64 TX_DFE_PORT0_CLN_CNT_HS_TRAIL Register (TX_DFE_PORT0_CLN_CNT_HS_TRAIL)—Offset FCh 5362
31.65 TX_DFE_PORT0_CLN_CNT_HS_EXIT Register (TX_DFE_PORT0_CLN_CNT_HS_EXIT)—Offset 100h..... 5362
31.66 TX_DFE_PORT0_CLN_CNT_LPX Register (TX_DFE_PORT0_CLN_CNT_LPX)—Offset 104h. 5362
31.67 TX_DFE_PORT0_CLN_CNT_PREP Register (TX_DFE_PORT0_CLN_CNT_PREP)—Offset 108h..... 5363
31.68 TX_DFE_PORT0_CLN_CNT_ZERO Register (TX_DFE_PORT0_CLN_CNT_ZERO)—Offset 10Ch..... 5363
31.69 TX_DFE_PORT0_CLN_CNT_PLL Register (TX_DFE_PORT0_CLN_CNT_PLL)—Offset 110h . 5364
31.70 TX_DFE_PORT0_CLN_CNT_POST Register (TX_DFE_PORT0_CLN_CNT_POST)—Offset 114h..... 5364
31.71 TX_DFE_PORT0_CLN_DLN_ENABLE_OVR Register (TX_DFE_PORT0_CLN_DLN_ENABLE_OVR)—Offset 118h..... 5364
31.72 TX_DFE_PORT0_RESERVED1 Register (TX_DFE_PORT0_RESERVED1)—Offset 11Ch 5365
31.73 TX_DFX_PORT1_CTL_REG Register (TX_DFX_PORT1_CTL_REG)—Offset 120h..... 5365
31.74 TX_DFX_PORT1_CD_COUNTUP_STATUS Register (TX_DFX_PORT1_CD_COUNTUP_STATUS)—Offset 124h 5366
31.75 TX_DFX_PORT1_RCC_CFG_REG Register (TX_DFX_PORT1_RCC_CFG_REG)—Offset 128h 5367
31.76 TX_DFX_PORT1_ACIO_CFG_REG Register (TX_DFX_PORT1_ACIO_CFG_REG)—Offset 12Ch..... 5367
31.77 TX_DFX_PORT1_ACIOSS_REG Register (TX_DFX_PORT1_ACIOSS_REG)—Offset 130h ... 5368
31.78 TX_DFX_PORT1_ACIOERR_REG Register (TX_DFX_PORT1_ACIOERR_REG)—Offset 134h 5369
31.79 TX_DFX_PORT1_DELAY_CALIB_REG Register (TX_DFX_PORT1_DELAY_CALIB_REG)—Offset 138h..... 5370
31.80 TX_DFX_PORT1_RXLPBK_DATA Register (TX_DFX_PORT1_RXLPBK_DATA)—Offset 13Ch 5370
31.81 TX_DFX_PORT1_CLANE_CTL_REG Register (TX_DFX_PORT1_CLANE_CTL_REG)—Offset 140h..... 5370
31.82 TX_DFX_PORT1_RESERVED1 Register (TX_DFX_PORT1_RESERVED1)—Offset 144h 5371
31.83 TX_DFX_PORT1_RESERVED2 Register (TX_DFX_PORT1_RESERVED2)—Offset 148h 5371
31.84 TX_DFX_PORT1_RESERVED3 Register (TX_DFX_PORT1_RESERVED3)—Offset 14Ch 5371
31.85 TX_DFX_PORT1_DLANE0_CTL_REG Register (TX_DFX_PORT1_DLANE0_CTL_REG)—Offset 150h..... 5372
31.86 TX_DFX_PORT1_DLANE0_STAT_REG Register (TX_DFX_PORT1_DLANE0_STAT_REG)—Offset 154h..... 5372
31.87 TX_DFX_PORT1_DLANE1_CTL_REG Register (TX_DFX_PORT1_DLANE1_CTL_REG)—Off-

set 158h 5373
31.88 TX_DFX_PORT1_DLANE1_STAT_REG Register (TX_DFX_PORT1_DLANE1_STAT_REG)—Offset 15Ch 5373
31.89 TX_DFX_PORT1_DLANE2_CTL_REG Register (TX_DFX_PORT1_DLANE2_CTL_REG)—Offset 160h 5374
31.90 TX_DFX_PORT1_DLANE2_STAT_REG Register (TX_DFX_PORT1_DLANE2_STAT_REG)—Offset 164h 5374
31.91 TX_DFX_PORT1_DLANE3_CTL_REG Register (TX_DFX_PORT1_DLANE3_CTL_REG)—Offset 168h 5375
31.92 TX_DFX_PORT1_DLANE3_STAT_REG Register (TX_DFX_PORT1_DLANE3_STAT_REG)—Offset 16Ch 5375
31.93 TX_DFX_PORT1_LANE_RESERVED5 Register (TX_DFX_PORT1_LANE_RESERVED5)—Offset 170h 5376
31.94 TX_DFX_PORT1_LANE_RESERVED6 Register (TX_DFX_PORT1_LANE_RESERVED6)—Offset 174h 5376
31.95 TX_DFX_PORT1_LANE_RESERVED7 Register (TX_DFX_PORT1_LANE_RESERVED7)—Offset 178h 5376
31.96 TX_DFX_PORT1_LANE_RESERVED8 Register (TX_DFX_PORT1_LANE_RESERVED8)—Offset 17Ch 5377
31.97 TX_DFE_PORT1_DLN_CNT_LPX Register (TX_DFE_PORT1_DLN_CNT_LPX)—Offset 180h 5377
31.98 TX_DFE_PORT1_DLN_CNT_HS_PREP Register (TX_DFE_PORT1_DLN_CNT_HS_PREP)—Offset 184h 5377
31.99 TX_DFE_PORT1_DLN_CNT_HS_ZERO Register (TX_DFE_PORT1_DLN_CNT_HS_ZERO)—Offset 188h 5378
31.100TX_DFE_PORT1_DLN_CNT_HS_TRAIL Register (TX_DFE_PORT1_DLN_CNT_HS_TRAIL)—Offset 18Ch 5378
31.101TX_DFE_PORT1_DLN_CNT_HS_EXIT Register (TX_DFE_PORT1_DLN_CNT_HS_EXIT)—Offset 190h 5379
31.102TX_DFE_PORT1_DLN_CNT_RX Register (TX_DFE_PORT1_DLN_CNT_RX)—Offset 194h 5379
31.103TX_DFE_PORT1_DLN_SYNC_CNT Register (TX_DFE_PORT1_DLN_SYNC_CNT)—Offset 198h 5379
31.104TX_DFE_PORT1_CLN_CNT_HS_TRAIL Register (TX_DFE_PORT1_CLN_CNT_HS_TRAIL)—Offset 19Ch 5380
31.105TX_DFE_PORT1_CLN_CNT_HS_EXIT Register (TX_DFE_PORT1_CLN_CNT_HS_EXIT)—Offset 1A0h 5380
31.106TX_DFE_PORT1_CLN_CNT_LPX Register (TX_DFE_PORT1_CLN_CNT_LPX)—Offset 1A4h 5380
31.107TX_DFE_PORT1_CLN_CNT_PREP Register (TX_DFE_PORT1_CLN_CNT_PREP)—Offset 1A8h 5381
31.108TX_DFE_PORT1_CLN_CNT_ZERO Register (TX_DFE_PORT1_CLN_CNT_ZERO)—Offset 1ACh 5381
31.109TX_DFE_PORT1_CLN_CNT_PLL Register (TX_DFE_PORT1_CLN_CNT_PLL)—Offset 1B0h 5382
31.110TX_DFE_PORT1_CLN_CNT_POST Register (TX_DFE_PORT1_CLN_CNT_POST)—Offset 1B4h 5382
31.111TX_DFE_PORT1_CLN_DLN_ENABLE_OVR Register (TX_DFE_PORT1_CLN_DLN_ENABLE_OVR)—Offset 1B8h 5382
31.112TX_DFE_PORT1_RESERVED1 Register (TX_DFE_PORT1_RESERVED1)—Offset 1BCh 5383
31.113RX_DWORD5 (rx_dword5)—Offset 4214h/4614h/1A14h/1614h/1214h/614h/614h/1E14h/4E14h/4A14h....5383

32 Display Registers 5385

32.1 LJPLL_RW_CONTROL_0 (LJPLL_CR_RW_CONTROL_0)—Offset 0h....5385
32.2 LJPLL_RW_CONTROL_1 (LJPLL_CR_RW_CONTROL_1)—Offset 10h 5385
32.3 LJPLL_RW_CONTROL_2 (LJPLL_CR_RW_CONTROL_2)—Offset 14h 5386

32.4 depll_cp (depll_cp)—Offset 20h 5386
32.5 depll_rac (depll_rac)—Offset 28h 5387
32.6 depll_wac (depll_wac)—Offset 30h 5387
32.7 MCHBAR_LSB (GFXVTDBAR_LSB)—Offset 6C88h 5387
32.8 MCHBAR_MSB (GFXVTDBAR_MSB)—Offset 6C8Ch 5388
32.9 Fuse Status (FUSE_STATUS)—Offset 42000h 5388
32.10 Display Engine Power 1 (DE_POWER1)—Offset 42400h 5389
32.11 Display Engine Power 2 (DE_POWER2)—Offset 42404h 5390
32.12 Master Interrupt Control (MASTER_INT_CTL)—Offset 44200h 5390
32.13 GT 0 Interrupts 0 (GT_0_INTERRUPT_0)—Offset 44300h 5392
32.14 GT 0 Interrupts 1 (GT_0_INTERRUPT_1)—Offset 44304h 5393
32.15 GT 0 Interrupts 2 (GT_0_INTERRUPT_2)—Offset 44308h 5395
32.16 GT 0 Interrupts 3 (GT_0_INTERRUPT_3)—Offset 4430Ch 5397
32.17 GT 1 Interrupts 0 (GT_1_INTERRUPT_0)—Offset 44310h 5399
32.18 GT 1 Interrupts 1 (GT_1_INTERRUPT_1)—Offset 44314h 5401
32.19 GT 1 Interrupts 2 (GT_1_INTERRUPT_2)—Offset 44318h 5402
32.20 GT 1 Interrupts 3 (GT_1_INTERRUPT_3)—Offset 4431Ch 5404
32.21 GT 2 Interrupts 0 (GT_2_INTERRUPT_0)—Offset 44320h 5406
32.22 GT 2 Interrupts 1 (GT_2_INTERRUPT_1)—Offset 44324h 5407
32.23 GT 2 Interrupts 2 (GT_2_INTERRUPT_2)—Offset 44328h 5408
32.24 GT 2 Interrupts 3 (GT_2_INTERRUPT_3)—Offset 4432Ch 5410
32.25 GT 3 Interrupts 0 (GT_3_INTERRUPT_0)—Offset 44330h 5411
32.26 GT 3 Interrupts 1 (GT_3_INTERRUPT_1)—Offset 44334h 5413
32.27 GT 3 Interrupts 2 (GT_3_INTERRUPT_2)—Offset 44338h 5415
32.28 GT 3 Interrupts 3 (GT_3_INTERRUPT_3)—Offset 4433Ch 5416
32.29 Display Engine Port Interrupts 0 (DE_PORT_INTERRUPT_0)—Offset 44440h 5418
32.30 Display Engine Port Interrupts 1 (DE_PORT_INTERRUPT_1)—Offset 44444h 5419
32.31 Display Engine Port Interrupts 2 (DE_PORT_INTERRUPT_2)—Offset 44448h 5421
32.32 Display Engine Port Interrupts 3 (DE_PORT_INTERRUPT_3)—Offset 4444Ch 5422
32.33 PCU Interrupts 0 (PCU_INTERRUPT_0)—Offset 444E0h 5423
32.34 PCU Interrupts 1 (PCU_INTERRUPT_1)—Offset 444E4h 5424
32.35 PCU Interrupts 2 (PCU_INTERRUPT_2)—Offset 444E8h 5424
32.36 PCU Interrupts 3 (PCU_INTERRUPT_3)—Offset 444ECh 5425
32.37 Display C State Enable (DC_STATE_EN)—Offset 45504h 5425
32.38 CD Clock Control (CDCLK_CTL)—Offset 46000h 5426
32.39 North Display Reset Warn Options (NDE_RSTWRN_OPT)—Offset 46408h 5427
32.40 Software Flags 0 (SWF_0)—Offset 4F000h 5428
32.41 Software Flags 1 (SWF_1)—Offset 4F004h 5428
32.42 Software Flags 2 (SWF_2)—Offset 4F008h 5428
32.43 Software Flags 3 (SWF_3)—Offset 4F00Ch 5429
32.44 Software Flags 4 (SWF_4)—Offset 4F010h 5429
32.45 Software Flags 5 (SWF_5)—Offset 4F014h 5429
32.46 Software Flags 6 (SWF_6)—Offset 4F018h 5430
32.47 Software Flags 7 (SWF_7)—Offset 4F01Ch 5430
32.48 Software Flags 8 (SWF_8)—Offset 4F020h 5430
32.49 Software Flags 9 (SWF_9)—Offset 4F024h 5431
32.50 Software Flags 10 (SWF_10)—Offset 4F028h 5431
32.51 Software Flags 11 (SWF_11)—Offset 4F02Ch 5431
32.52 Software Flags 12 (SWF_12)—Offset 4F030h 5432
32.53 Software Flags 13 (SWF_13)—Offset 4F034h 5432
32.54 Software Flags 14 (SWF_14)—Offset 4F038h 5432
32.55 Software Flags 15 (SWF_15)—Offset 4F03Ch 5433
32.56 Software Flags 16 (SWF_16)—Offset 4F040h 5433
32.57 Software Flags 17 (SWF_17)—Offset 4F044h 5433
32.58 Software Flags 18 (SWF_18)—Offset 4F048h 5434

32.59 Software Flags 19 (SWF_19)—Offset 4F04Ch 5434
32.60 Software Flags 20 (SWF_20)—Offset 4F050h 5434
32.61 Software Flags 21 (SWF_21)—Offset 4F054h 5435
32.62 Software Flags 22 (SWF_22)—Offset 4F058h 5435
32.63 Software Flags 23 (SWF_23)—Offset 4F05Ch 5435
32.64 Software Flags 24 (SWF_24)—Offset 4F060h 5436
32.65 Software Flags 25 (SWF_25)—Offset 4F064h 5436
32.66 Software Flags 26 (SWF_26)—Offset 4F068h 5436
32.67 Software Flags 27 (SWF_27)—Offset 4F06Ch 5437
32.68 Software Flags 28 (SWF_28)—Offset 4F070h 5437
32.69 Software Flags 29 (SWF_29)—Offset 4F074h 5437
32.70 Software Flags 30 (SWF_30)—Offset 4F078h 5438
32.71 Software Flags 31 (SWF_31)—Offset 4F07Ch 5438
32.72 Software Flags 32 (SWF_32)—Offset 4F080h 5438
32.73 Software Flags 33 (SWF_33)—Offset 4F084h 5439
32.74 Software Flags 34 (SWF_34)—Offset 4F088h 5439
32.75 Software Flags 35 (SWF_35)—Offset 4F08Ch 5439
32.76 Display Fuse (DFSM)—Offset 51000h 5440
32.77 Display Strap State (DSSM)—Offset 51004h 5441
32.78 Display Fuse Done (DFSDONE)—Offset 51080h 5443
32.79 DDI A Buffer Control (DDI_BUF_CTL_A)—Offset 64000h 5443
32.80 DDI B Buffer Control (DDI_BUF_CTL_B)—Offset 64100h 5444
32.81 DDI C Buffer Control (DDI_BUF_CTL_C)—Offset 64200h 5445
32.82 DDI D Buffer Control (DDI_BUF_CTL_D)—Offset 64300h 5447
32.83 DDI E Buffer Control (DDI_BUF_CTL_E)—Offset 64400h 5448
32.84 DDI F Buffer Control (DDI_BUF_CTL_F)—Offset 64500h 5449
32.85 Hot Plug Control (HOTPLUG_CTL)—Offset C4030h 5450
32.86 HPD Pulse Count DDI B (HPD_PULSE_CNT_B)—Offset C4034h 5451
32.87 HPD Filter Count (HPD_FILTER_CNT)—Offset C4038h 5451
32.88 HPD Pulse Count DDI C (HPD_PULSE_CNT_C)—Offset C4044h 5452
32.89 HPD Pulse Count DDI A (HPD_PULSE_CNT_A)—Offset C404Ch 5452
32.90 FENCE0_LSB (FENCE0_LSB)—Offset 100000h 5452
32.91 FENCE0_MSB (FENCE0_MSB)—Offset 100004h 5453
32.92 FENCE1_LSB (FENCE1_LSB)—Offset 100008h 5453
32.93 FENCE1_MSB (FENCE1_MSB)—Offset 10000Ch 5454
32.94 FENCE2_LSB (FENCE2_LSB)—Offset 100010h 5454
32.95 FENCE2_MSB (FENCE2_MSB)—Offset 100014h 5455
32.96 FENCE3_LSB (FENCE3_LSB)—Offset 100018h 5455
32.97 FENCE3_MSB (FENCE3_MSB)—Offset 10001Ch 5456
32.98 FENCE4_LSB (FENCE4_LSB)—Offset 100020h 5456
32.99 FENCE4_MSB (FENCE4_MSB)—Offset 100024h 5457
32.100FENCE5_LSB (FENCE5_LSB)—Offset 100028h 5457
32.101FENCE5_MSB (FENCE5_MSB)—Offset 10002Ch 5458
32.102FENCE6_LSB (FENCE6_LSB)—Offset 100030h 5458
32.103FENCE6_MSB (FENCE6_MSB)—Offset 100034h 5459
32.104FENCE7_LSB (FENCE7_LSB)—Offset 100038h 5459
32.105FENCE7_MSB (FENCE7_MSB)—Offset 10003Ch 5460
32.106FENCE8_LSB (FENCE8_LSB)—Offset 100040h 5460
32.107FENCE8_MSB (FENCE8_MSB)—Offset 100044h 5461
32.108FENCE9_LSB (FENCE9_LSB)—Offset 100048h 5461
32.109FENCE9_MSB (FENCE9_MSB)—Offset 10004Ch 5462
32.110FENCE10_LSB (FENCE10_LSB)—Offset 100050h 5462
32.111FENCE10_MSB (FENCE10_MSB)—Offset 100054h 5463
32.112FENCE11_LSB (FENCE11_LSB)—Offset 100058h 5463
32.113FENCE11_MSB (FENCE11_MSB)—Offset 10005Ch 5464

32.114FENCE12_LSB (FENCE12_LSB)—Offset 100060h 5464
32.115FENCE12_MSB (FENCE12_MSB)—Offset 100064h 5465
32.116FENCE13_LSB (FENCE13_LSB)—Offset 100068h 5465
32.117FENCE13_MSB (FENCE13_MSB)—Offset 10006Ch 5466
32.118FENCE14_LSB (FENCE14_LSB)—Offset 100070h 5466
32.119FENCE14_MSB (FENCE14_MSB)—Offset 100074h 5467
32.120FENCE15_LSB (FENCE15_LSB)—Offset 100078h 5467
32.121FENCE15_MSB (FENCE15_MSB)—Offset 10007Ch 5468
32.122FENCE16_LSB (FENCE16_LSB)—Offset 100080h 5468
32.123FENCE16_MSB (FENCE16_MSB)—Offset 100084h 5469
32.124FENCE17_LSB (FENCE17_LSB)—Offset 100088h 5469
32.125FENCE17_MSB (FENCE17_MSB)—Offset 10008Ch 5470
32.126FENCE18_LSB (FENCE18_LSB)—Offset 100090h 5470
32.127FENCE18_MSB (FENCE18_MSB)—Offset 100094h 5471
32.128FENCE19_LSB (FENCE19_LSB)—Offset 100098h 5471
32.129FENCE19_MSB (FENCE19_MSB)—Offset 10009Ch 5472
32.130FENCE20_LSB (FENCE20_LSB)—Offset 1000A0h 5472
32.131FENCE20_MSB (FENCE20_MSB)—Offset 1000A4h 5473
32.132FENCE21_LSB (FENCE21_LSB)—Offset 1000A8h 5473
32.133FENCE21_MSB (FENCE21_MSB)—Offset 1000ACh 5474
32.134FENCE22_LSB (FENCE22_LSB)—Offset 1000B0h 5474
32.135FENCE22_MSB (FENCE22_MSB)—Offset 1000B4h 5475
32.136FENCE23_LSB (FENCE23_LSB)—Offset 1000B8h 5475
32.137FENCE23_MSB (FENCE23_MSB)—Offset 1000BCh 5476
32.138FENCE24_LSB (FENCE24_LSB)—Offset 1000C0h 5476
32.139FENCE24_MSB (FENCE24_MSB)—Offset 1000C4h 5477
32.140FENCE25_LSB (FENCE25_LSB)—Offset 1000C8h 5477
32.141FENCE25_MSB (FENCE25_MSB)—Offset 1000CCh 5478
32.142FENCE26_LSB (FENCE26_LSB)—Offset 1000D0h 5478
32.143FENCE26_MSB (FENCE26_MSB)—Offset 1000D4h 5479
32.144FENCE27_LSB (FENCE27_LSB)—Offset 1000D8h 5479
32.145FENCE27_MSB (FENCE27_MSB)—Offset 1000DCh 5480
32.146FENCE28_LSB (FENCE28_LSB)—Offset 1000E0h 5480
32.147FENCE28_MSB (FENCE28_MSB)—Offset 1000E4h 5481
32.148FENCE29_LSB (FENCE29_LSB)—Offset 1000E8h 5481
32.149FENCE29_MSB (FENCE29_MSB)—Offset 1000ECh 5482
32.150FENCE30_LSB (FENCE30_LSB)—Offset 1000F0h 5482
32.151FENCE30_MSB (FENCE30_MSB)—Offset 1000F4h 5483
32.152FENCE31_LSB (FENCE31_LSB)—Offset 1000F8h 5483
32.153FENCE31_MSB (FENCE31_MSB)—Offset 1000FCh 5484
32.154DPFC_CONTROL_SA (DPFC_CTL_SA)—Offset 100100h 5484
32.155DPFC_CPU_FENCE_OFFSET (DPFC_CFO)—Offset 100104h 5485
32.156TILECTL (TILECTL)—Offset 101000h 5485
32.157GFX_FLSH_CNT (GFX_FLSH_CNT)—Offset 101008h 5486
32.158DISPLAY_DEADLINE_CONTROL (DISPLAY_DEADLINE_CONTROL)—Offset 10102Ch 5486
32.159DG_CLKREQ_POLICY (DG_CLKREQ_POLICY)—Offset 101038h 5487
32.160VDMBDFBARKVM (VDMBDFBARKVM)—Offset 101070h 5487
32.161WDBDF (WDBDF)—Offset 101074h 5488
32.162TOUCHBDF (TOUCHBDF)—Offset 101078h 5488
32.163MTOLUD (MTOLUD)—Offset 108000h 5489
32.164MMIO Mirror of GMCH Graphics Control Register (MGGC)—Offset 108040h 5490
32.165MTOUUD_LSB (MTOUUD_LSB)—Offset 108080h 5491
32.166MTOUUD_MSB (MTOUUD_MSB)—Offset 108084h 5491
32.167MBDSM (MBDSM)—Offset 1080C0h 5492
32.168MBGSM (MBGSM)—Offset 108100h 5492

32.169Base of DMA Protected Range (BDPR)—Offset 108140h 5493
32.170MPMEN (MPMEN)—Offset 108180h 5493
32.171MPLMBASE (MPLMBASE)—Offset 1081C0h.....5494
32.172MPLMLIMIT (MPLMLIMIT)—Offset 108200h.....5495
32.173MPHMBASE_LSB (MPHMBASE_LSB)—Offset 108240h.....5495
32.174MPHMBASE_MSB (MPHMBASE_MSB)—Offset 108244h 5496
32.175MPHMLIMIT_LSB (MPHMLIMIT_LSB)—Offset 108280h.....5496
32.176MPHMLIMIT_MSB (MPHMLIMIT_MSB)—Offset 108284h 5497
32.177MGCMD (MGCMD)—Offset 108300h.....5497
32.178MEMRR_BASE_LSB (MEMRR_BASE_LSB)—Offset 108340h 5498
32.179MEMRR_BASE_MSB (MEMRR_BASE_MSB)—Offset 108344h.....5498
32.180MEMRR_MASK_LSB (MEMRR_MASK_LSB)—Offset 108380h 5499
32.181MEMRR_MASK_MSB (MEMRR_MASK_MSB)—Offset 108384h.....5499
32.182GTACK (GTACK)—Offset 120004h 5500
32.183EDRAMCAP (EDRAMCAP)—Offset 120010h 5500
32.184MAILBOX0 (MAILBOX0)—Offset 120800h 5501
32.185MAILBOX1 (MAILBOX1)—Offset 120804h 5501
32.186MAILBOX2 (MAILBOX2)—Offset 120808h 5501
32.187MAILBOX3 (MAILBOX3)—Offset 12080Ch 5502
32.188FLT_RPT0 (FLT_RPT0)—Offset 124810h 5502
32.189FLT_RPT1 (FLT_RPT1)—Offset 124814h 5502
32.190FLT_RPT2 (FLT_RPT2)—Offset 124818h 5503
32.191FLT_RPT3 (FLT_RPT3)—Offset 12481Ch 5503
32.192PPRO (PPRO)—Offset 124820h 5504
32.193PPPR (PPPR)—Offset 124824h 5504
32.194RTADDR_LSB (RTADDR_LSB)—Offset 124830h.....5504
32.195RTADDR_MSB (RTADDR_MSB)—Offset 124834h 5505
32.196GT Scratchpad 0 (GTSP0)—Offset 130040h 5505
32.197GT Scratchpad 1 (GTSP1)—Offset 130044h 5506
32.198GT Scratchpad 2 (GTSP2)—Offset 130048h 5506
32.199GT Scratchpad 3 (GTSP3)—Offset 13004Ch 5506
32.200GT Scratchpad 4 (GTSP4)—Offset 130050h 5507
32.201GT Scratchpad 5 (GTSP5)—Offset 130054h 5507
32.202GT Scratchpad 6 (GTSP6)—Offset 130058h 5507
32.203GT Scratchpad 7 (GTSP7)—Offset 13005Ch 5507
32.204Version Register (VER_REG_0_0_0_VTDBAR)—Offset 0h.....5508
32.205Capability Register (CAP_REG_0_0_0_VTDBAR)—Offset 8h 5508
32.206Extended Capability Register (ECAP_REG_0_0_0_VTDBAR)—Offset 10h 5511
32.207Global Command Register (GCMD_REG_0_0_0_VTDBAR)—Offset 18h.....5514
32.208Global Status Register (GSTS_REG_0_0_0_VTDBAR)—Offset 1Ch 5516
32.209Root Table Address Register (RTADDR_REG_0_0_0_VTDBAR)—Offset 20h 5517
32.210Context Command Register (CCMD_REG_0_0_0_VTDBAR)—Offset 28h.....5518
32.211Fault Status Register (FSTS_REG_0_0_0_VTDBAR)—Offset 34h 5519
32.212Fault Event Control Register (FECTL_REG_0_0_0_VTDBAR)—Offset 38h.....5520
32.213LJPLL_RW_CONTROL_0 (LJPLL_CR_RW_CONTROL_0)—Offset 0h 5521
32.214LJPLL_RW_CONTROL_1 (LJPLL_CR_RW_CONTROL_1)—Offset 10h 5522
32.215LJPLL_RW_CONTROL_2 (LJPLL_CR_RW_CONTROL_2)—Offset 14h 5523
32.216dsipll_cp (dsipll_cp)—Offset 20h 5523
32.217dsipll_rac (dsipll_rac)—Offset 28h.....5523
32.218dsipll_wac (dsipll_wac)—Offset 30h.....5524
32.219LJPLL_RW_CONTROL_0 (LJPLL_CR_RW_CONTROL_0)—Offset 1000h 5524
32.220LJPLL_RW_CONTROL_1 (LJPLL_CR_RW_CONTROL_1)—Offset 1010h 5525
32.221LJPLL_RW_CONTROL_2 (LJPLL_CR_RW_CONTROL_2)—Offset 1014h 5525
32.222Capability Register (CAP_REG_0_0_0_VTDBAR)—Offset 8h 5526
32.223Extended Capability Register (ECAP_REG_0_0_0_VTDBAR)—Offset 10h 5529

32.224 Global Command Register (GCMD_REG_0_0_0_VTDBAR)—Offset 18h 5532
32.225 Global Status Register (GSTS_REG_0_0_0_VTDBAR)—Offset 1Ch 5534
32.226 Root Table Address Register (RTADDR_REG_0_0_0_VTDBAR)—Offset 20h 5536
32.227 Context Command Register (CCMD_REG_0_0_0_VTDBAR)—Offset 28h 5536
32.228 Fault Status Register (FSTS_REG_0_0_0_VTDBAR)—Offset 34h 5538
32.229 Fault Event Control Register (FECTL_REG_0_0_0_VTDBAR)—Offset 38h 5539

33 USB 2 Tuning Registers 5541
33.1 c73usb280_USB2 PER PORT (USB2_PER_PORT_PP0)—Offset 4100h 5541
33.2 c73usb280_USB2 PER PORT 2 (USB2_PER_PORT_2_PP0)—Offset 4126h 5543
33.3 c73usb280_USB2 PER PORT (USB2_PER_PORT_PP1)—Offset 4200h 5545
33.4 c73usb280_USB2 PER PORT 2 (USB2_PER_PORT_2_PP1)—Offset 4226h 5547
33.5 c73usb280_USB2 PER PORT (USB2_PER_PORT_PP2)—Offset 4300h 5549
33.6 c73usb280_USB2 PER PORT 2 (USB2_PER_PORT_2_PP2)—Offset 4326h 5552
33.7 c73usb280_USB2 PER PORT (USB2_PER_PORT_PP3)—Offset 4400h 5554
33.8 c73usb280_USB2 PER PORT 2 (USB2_PER_PORT_2_PP3)—Offset 4426h 5556
33.9 c73usb280_USB2 PER PORT (USB2_PER_PORT_PP4)—Offset 4500h 5558
33.10 c73usb280_USB2 PER PORT 2 (USB2_PER_PORT_2_PP4)—Offset 4526h 5560
33.11 c73usb280_USB2 PER PORT (USB2_PER_PORT_PP5)—Offset 4600h 5562
33.12 c73usb280_USB2 PER PORT 2 (USB2_PER_PORT_2_PP5)—Offset 4626h 5564
33.13 c73usb280_USB2 PER PORT (USB2_PER_PORT_PP6)—Offset 4700h 5566
33.14 c73usb280_USB2 PER PORT 2 (USB2_PER_PORT_2_PP6)—Offset 4726h 5569
33.15 c73usb280_USB2 PER PORT (USB2_PER_PORT_PP7)—Offset 4800h 5571
33.16 c73usb280_USB2 PER PORT 2 (USB2_PER_PORT_2_PP7)—Offset 4826h 5573
33.17 c73usb280_USB2 COMPBG (USB2_COMPBG)—Offset 7F04h 5575

34 Real Time Clock (RTC) Registers 5578
34.1 RTC Index Register (INDEX)—Offset 70h 5578
34.2 RTC Target Register (TARGET)—Offset 71h 5578
34.3 Seconds (Sec)—Offset 0h 5578
34.4 Seconds Alarm (Sec_Alarm)—Offset 1h 5579
34.5 Minutes (Minutes)—Offset 2h 5579
34.6 Minutes Alarm (Minutes_Alarm)—Offset 3h 5579
34.7 Hours (Hours)—Offset 4h 5580
34.8 Hours Alarm (Hours_Alarm)—Offset 5h 5580
34.9 Day of Week (Day_of_Week)—Offset 6h 5580
34.10 Day of Month (Day_of_Month)—Offset 7h 5581
34.11 Month (Month)—Offset 8h 5581
34.12 Year (Year)—Offset 9h 5581
34.13 Register A (Register_A)—Offset Ah 5582
34.14 Register B - General Configuration (Register_B)—Offset Bh 5582
34.15 Register C - Flag Register (Register_C)—Offset Ch 5583
34.16 Register D - Flag Register (Register_D)—Offset Dh 5584
34.17 114 Bytes of Lower User RAM (Register_E)—Offset Eh 5584
34.18 128 Bytes of Upper User RAM (Register_80)—Offset 80h 5585
34.19 RTC Configuration (RC)—Offset 3400h 5585
34.20 Backed Up Control (BUC)—Offset 3414h 5586
34.21 RTC Dynamic Clock Gating Control (RTCDCG)—Offset 3418h 5587

Revision History

Document Number	Revision Number	Description	Revision Date
336561	001	Initial Release	February 2018

§ §

1 Introduction

This is Volume 2 of the Intel® Pentium® Silver and Intel® Celeron® Processors Datasheet. Volume 2 provides register information for the SoC.

Refer to document #336560 for the Intel® Pentium® Silver and Intel® Celeron® Processors Datasheet, Volume 1.

Throughout this document Intel® Pentium® Silver and Intel® Celeron® Processors is referred as SoC.

Throughout this document Intel® Pentium® Silver and Intel® Celeron® Processors families refer to:

- Intel® Pentium® Silver N5000
- Intel® Pentium® Silver J5005
- Intel® Celeron® N4000 and N4100
- Intel® Celeron® J4105 and J4005

§ §

2 Host Bridge/DRAM Registers

This chapter documents the registers in Bus: 0, Device 0, Function 0.

NOTE: These registers apply to all processors.

2.1 Device ID and Vendor ID Register (DEVICE_ID_VENDOR_ID_0_0_0_PCI) — Offset 0h

This register uniquely identifies any PCI device.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + 0h	AF08086 h

Bit Range	Default & Access	Field Name (ID): Description
31:23	15h RW/V	**Strap2 Portion of Device ID (DEVICE_ID_STRAP2):** The reset default value of this register is tied to the parameter DEVICE_ID_STRAP2. These bits can be rewritten from SETIDVALUE message 1st DW data byte2, byte3.
22:21	3h RO	**Strap1 Portion of Device ID (DEVICE_ID_STRAP1):** These bits are tied to a C-Unit parameter DEVICE_ID_STRAP1.
20:16	10h RO/V	**Fuse Backed Portion of Device ID (DEVICE_ID_FUSE):** These bits of Device ID are fuse backed.
15:0	8086h RO	**Vendor ID (VENDOR_ID):** Hardwired to Intel's Vendor ID value.

2.2 PCI Status and PCI Command Register (PCI_STATUS_COMMAND_0_0_0_PCI) — Offset 4h

PCI Status is used to record status information for PCI bus related events. PCI Command provides coarse control over a device's ability to generate and respond to PCI cycles.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + 4h	7 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	7h RO	**PCI Status and PCI Command (PCI_STATUS_AND_COMMAND):** Hardwired to 32'h00000007, allowing this device to respond to I/O space and Memory Space accesses. This device is also allowed to behave as a bus master.

2.3 PCI Revision ID and PCI Class Code Register (REVISION_ID_CLASS_CODE_0_0_0_PCI) — Offset 8h

Revision ID contains the revision number of the device. Class Code identifies the basic function of the device.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + 8h	6000000 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	600h RO	**Class Code 1 (CLASS_CODE1):** Hardwired 16'h0600, indicating this device is a Host Bridge.
15:8	0h RO	**Class Code 0 (CLASS_CODE0):** Hardwired to 8'h00, indicating this device is a Host Bridge.
7:0	0h RW/V	**Revision ID Strap (REVISION_ID_STRAP):** These bits can be rewritten from SETIDVALUE message 1st DW data byte 0.

2.4 Master Latency Timer and Header Type Register (MASTER_LATENCY_TIME_0_0_0_PCI) — Offset Ch

This register defines Latency Timer and layout of the device Configuration Space header.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + Ch	800000 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RESERVED_1):** Reserved
23:16	80h RO	**Header Type (HEADER_TYPE):** This field identifies the layout of the second part of the predefined header (beginning at byte 10h in the Configuration Space) and also whether or not the device contains multiple functions. Hardwired to 8'h80, indicating this is a multi-function device.
15:8	0h RO	**Master Latency Timer (MASTER_LATENCY_TIMER):** This field specifies, in units of PCI bus clocks, the value of the Latency Timer for this PCI bus master. Hardwired to 1'b0.
7:0	0h RO	**Reserved (RESERVED_0):** Reserved

2.5 System Agent Memory Mapped Range Base Address Register (SABAR) — Offset 10h

This register defines the base address for the System Agent Memory Mapped Configuration space. There is no physical memory within this 8KB window that can be addressed. The 8KB reserved by this register does not alias to any PCI 2.3 compliant memory mapped space. On reset, the System Agent Memory Mapped Configuration space is disabled and must be enabled by writing a 1 to the SABAREN bit.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + 10h	0 h

Bit Range	Default & Access	Field Name (ID): Description
Reserved		

2.6 PCI Subsystem Vendor ID and PCI Subsystem ID (SVID_SID_0_0_0_PCI) — Offset 2Ch

This register is used to uniquely identify the subsystem where the PCI device resides.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Subsystem ID (SUBSYSTEM_ID):** PCI Subsystem ID: This field should be programmed during BIOS initialization.
15:0	0h RW	**Subsystem Vendor ID (SUBSYSTEM_VENDOR_ID):** PCI Subsystem Vendor ID: This field should be programmed by BIOS during bootup to indicate the vendor of the system board.

2.7 Capability Register Pointer (CAPPTR_0_0_0_PCI) — Offset 34h

This register contains a pointer to the first Capability Register in a linked list.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + 34h	E0 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RESERVED_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
7:0	E0h RO	**Base Address of First Capability Register (BASE_ADDR):** This pointer is an 8-bit address to an offset within this device's Configuration Space that holds the first Capability Register (CAPID0_CAPCTRL).

2.8 B-Unit Copy of the MCHBAR (B_CR_MCHBAR_LO_0_0_0_PCI) — Offset 48h

This register contains the lower 32bits of the MCHBAR.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + 48h	FED10001 h

Bit Range	Default & Access	Field Name (ID): Description
31:15	1FDA2h RW	**Base Address of MCHBAR (BASE_ADDR):** Defines the base address of the MCHBAR. MCHBAR[38:15] is {MCHBAR_HI[6:0],MCHBAR_LO[31:15]}. If incoming Request Address[38:15] matches MCHBAR[38:15] then request hits an address in the MCHBAR range.
14:4	0h RO	**Base Address Not Implemented (BASE_ADDR_NOT_IMPLEMENTED):** Hardwired to 0 to indicate size of BAR 32kB.
3:1	0h RO	**Reserved (RESERVED_0):** Reserved
0	1h RW	**MCHBAR Enable (MCHBAREN):** • 0: MCHBAR is disabled and does not claim any memory • 1: MCHBAR memory mapped accesses are claimed and decoded appropriately

2.9 B-Unit Copy of the MCHBAR (B_CR_MCHBAR_HI_0_0_0_PCI) — Offset 4Ch

This register contains the upper 32 bits of the MCHBAR.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + 4Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Reserved (RESERVED_0):** Reserved
6:0	0h RW	**Base Address of MCHBAR (BASE_ADDR):** Defines the base address of the MCHBAR. If incoming Request Address[38:15] matches MCHBAR[38:15], then request hits an address in the MCHBAR range.

2.10 B-Unit Shadow of GGC (B_CR_GGC_0_0_0_PCI) — Offset 50h

B-Unit shadow of the GMCH Graphics Control Register.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:0, F:0] + 50h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RW	**Graphics Memory Select (GMS):** This field is used to select the amount of Main Memory that is preallocated to support the Internal Graphics device in VGA nonlinear and Native linear modes. The BIOS ensures that memory is preallocated only when Internal graphics is enabled. This register is also Intel TXT lockable. Hardware does not clear or set any of these bits automatically based on IGD being disabled/enabled. BIOS Requirement: BIOS must not set this field to 0h if IVD bit 1 of this register is 0. • 00h:0MB • 01h:32MB • 02h:64MB • 03h:96MB • 04h:128MB • 05h:160MB • 06h:192MB • 07h:224MB • 08h:256MB • 09h:288MB • 0Ah:320MB • 0Bh:352MB • 0Ch:384MB • 0Dh:416MB • 0Eh:448MB • 0Fh:480MB • 10h:512MB • 20h:...1024MB... • 30h:...1536MB... • 3Fh:...2016MB • 40h-FFh:Illegal value
7:6	0h RW	**Size of Graphics Translation Table Memory (GGMS):** This field is used to select the amount of Main Memory that is preallocated to support the Internal Graphics Translation Table. The BIOS ensures that memory is preallocated only when Internal graphics is enabled. GSM is assumed to be a contiguous physical DRAM space with DSM and BIOS needs to allocate a contiguous memory chunk. Hardware will derive the base of GSM from DSM only using the GSM size programmed in the register. Hardware functionality in case of programming this value to Reserved is not guaranteed. 0x0:No Preallocated Memory 0x1:2MB of Preallocated Memory 0x2:4MB of Preallocated Memory 0x3:8MB of Preallocated Memory
5:3	0h RO	**Reserved (RESERVED_0):** Reserved
2	0h RW	**Versatile Acceleration Mode Enable (VAMEN):** Enables the use of the iGFX engines for Versatile Acceleration. • 1: iGFX engines are in Versatile Acceleration Mode. Device 2 Class Code is 048000h. • 0: iGFX engines are in iGFX Mode. Device 2 Class Code is 030000h.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW/L	**IGA VGD Disable (IVD):** • 0: Enable. Device 2 IGD claims VGA memory and IO cycles, and the SubClass Code within Device 2 Class Code register is 00. • 1: Disable. Device 2 IGD does not claim VGA cycles Mem and IO, and the Sub Class Code field within Device 2 function 0 Class Code register is 80. BIOS Requirement: BIOS must not set this bit to 0 if the GMS field bits 7:3 of this register preallocates no memory. This bit MUST be set to 1 if Device 2 is disabled either via a fuse or fuse override B_CR_CAPID0_A[IGD] 1 or via a register DEVEN[3] 0. This register is locked by Intel TXT lock.
0	0h RW	**GGCLCK:** Reserved. Unused by the B-Unit.

2.11 B-Unit Shadow of the DEVEN Register (B_CR_DEVEN_0_0_0_PCI) — Offset 54h

The DEVEN register allows for enabling/disabling of PCI devices and functions that are within the CPU package.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + 54h	13 h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RO	**Reserved (RESERVED_1):** Reserved
5	0h RW	**Device 3 Function 0 Enable (D3F0EN):** • 0: Bus 0 Device 3 Function 0 is disabled and hidden • 1: Bus 0 Device 3 Function 0 is enabled and visible This bit will be set to 0b and remain 0b if Device 3 capability is disabled.
4	1h RW	**Device 2 Function 0 Enable (D2F0EN):** • 0: Bus 0 Device 2 Function 0 is disabled and hidden • 1: Bus 0 Device 2 Function 0 is enabled and visible This bit will be set to 0b and remain 0b if Device 2 capability is disabled.
3:2	0h RO	**Reserved (RESERVED_0):** Reserved
1	1h RW	**Device 0 Function 1 Enable (D0F1EN):** • 0: Bus 0 Device 1 Function 0 is disabled and hidden • 1: Bus 0 Device 1 Function 0 is enabled and visible This bit will be set to 0b and remain 0b if Device 0/0/1 capability is disabled.
0	1h RO	**Device 0 Function 0 Enable (D0F0EN):** Bus 0 Device 0 Function 0 may not be disabled, and is therefore hardwired to 1.

2.12 Protected Audio Video Path Control (PAVPC_0_0_0_PCI) — Offset 58h

This register contains the control bits for the Protected Audio Video Path.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + 58h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW	**Base of Protected Content Memory (PCMBASE):** 1M, 2M, 4M, and 8M. Base value programmed from Top of Stolen Memory itself defines the size of the Write Only Protected Content Memory (WOPCM). Separate WOPCM size programming is redundant information and not required. Default 1M size programming.
19:7	0h RO	**Reserved (RSVD2):** Reserved
6	0h RW	**ASMF Enable (ASMFEN):** ASMF method enable • 0: Disabled (default) • 1: Enabled
5	0h RO	**Reserved (RSVD1):** Reserved
4	0h RW	**Override Attack (OVTATTACK):** Override of Unsolicited Connection State Attack and Terminate. • 0: Disable Override. Attack Terminate allowed. • 1: Enable Override. Attack Terminate disallowed.
3	0h RW	**Heavy Mode Select (HVYMODSEL):** This bit is applicable only for PAVP2 operation mode with a bit also set or for PAVP3 mode only if the perApp memory config is disabled due to the clearing of an additional bit 9 in the Crypto Function Control_1 register address 0x320F0. • 0: Lite Mode NonSerpent mode • 1: Serpent Mode Note that PAVP2 or PAVP3 mode selection is done by programming bit 8 of the MFX_MODE Video Mode register.
2	0h RW	**PAVP Register Lock (PAVPLCK):** This lock bit only impacts the Display copy of this register. In the C-Unit, all register protection is implemented with SAI policy groups. This bit is maintained in the C-Unit for software observability. Display description: This bit locks all writeable contents in this register when set, including itself. Only a hardware reset can unlock the register again. This lock bit needs to be set only if PAVP is enabled (bit 1 of this register is asserted.)
1	0h RW	**Protected Audio Video Path Enable (PAVPE):** • 0: PAVP functionality is disabled. • 1: PAVP functionality is enabled.
0	0h RW	**Protected Content Memory Enable (PCME):** This field enables Protected Content Memory within Graphics Stolen Memory. This memory is the same as the Write Only Protected Content Memory area whose size is defined by bit 5 of this register. This register is locked when PAVPLCK is set. • 0: Protected Content Memory is disabled and cannot be programmed in this manner when PAVP is enabled. • 1: Protected Content Memory is enabled and is the only programming option available when PAVP is enabled. For non-PAVP3 Mode even for Lite mode configuration this bit should be programmed to 1'b1 and HVYMODESEL to 1'b0. This bit should always be programmed to 1'b1 if bits 1 and 2 (PAVPE and PAVP lock bits) are both set. With perApp Memory configuration support the range check for the Write Only Protected Content Memory area should always happen when this bit is set regardless of Lite or Serpent mode or PAVP2 or PAVP3 mode programming.

2.13 B-Unit PCI Express Enhanced Configuration Range Base Address Low (B_CR_PCIEXBAR_LO_0_0_0_PCI) — Offset 60h

Defines the base address of the PCI Express Enhanced Configuration region. This register contains the lower 32bits of PCIEXBAR.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + 60h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**PCI Express Base Address (PCIEXBAR):** PCIEXBAR[38:28] is {PCIEXBAR_HI[6:0],PCIEXBAR_LO[31:28]}. Describes bits [38:28] of the base address of the contiguous 256MB region for PCI Express Enhanced Configuration region. If bits [38:28] of the request address matches the PCIEXBAR[38:28] then the request targets the PCI Express Enhanced Configuration Space region. A posted memory operation from an IDI agent will be treated as a nonposted operation by the T-Unit and A-Unit.
27	0h RO	**ADMSK128:** Reserved. Unused by SoC since PCIEXBAR is a fixed 256MB region in SoCs.
26	0h RO	**ADMSK64:** Reserved. Unused by SoC since PCIEXBAR is a fixed 256MB region in SoCs.
25:3	0h RO	**Reserved (RESERVED_1):** Reserved
2:1	0h RO	**LENGTH:** Reserved and set to 0 indicating a fixed 256 MB region.
0	0h RW	**PCIEXBAR Range Enable (PCIEXBAREN):** • 0: PCIEXBAR range is disabled. Address may target DRAM or MMIO, depending on other address decode rules. • 1: PCIEXBAR range is enabled. Incoming request address must be compared with PCIEXBAR to determine whether the request targets PCI Express Enhanced Configuration region.

2.14 B-Unit PCI Express Enhanced Configuration Range Base Address High (B_CR_PCIEXBAR_HI_0_0_0_PCI) — Offset 64h

Defines part of the base address of the PCI Express Enhanced Configuration region. See bit field description.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + 64h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Reserved (RESERVED_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
6:0	0h RW	**PCI Express Base Address (PCIEXBAR):** PCIEXBAR[38:28] is {PCIEXBAR_HI[6:0],PCIEXBAR_LO[31:28]}. Describes bits [38:28] of the base address of the contiguous 256MB region for PCI Express Enhanced Configuration region. If bits [38:28] of the request address matches the PCIEXBAR[38:28] then the request targets the PCI Express Enhanced Configuration Space region. A posted memory operation from an IDI agent will be treated as a nonposted operation by the T-unit and A-unit.

2.15 Message Signaled Interrupts Capability Identifier and Message Control (MSI_CAPID) — Offset 90h

This register reflects the capability list of the system agent.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + 90h	0 h

Bit Range	Default & Access	Field Name (ID): Description
Reserved		

2.16 Message Address (MA) — Offset 94h

This register is used by firmware or software to assign an MSI address to the device.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + 94h	0 h

Bit Range	Default & Access	Field Name (ID): Description
Reserved		

2.17 Message Data (MD) — Offset 98h

This register is used by firmware/software to assign an MSI data to the device.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + 98h	0 h

Bit Range	Default & Access	Field Name (ID): Description
Reserved		

2.18 Error Status Register (ERRSTS) — Offset A0h

This register is used to report various error conditions. An MSI message is generated when any of these flags is set (if enabled by the ERRCMD and PCICMDSTS registers). These bits are set regardless of whether or not the MSI is enabled and generated. This register is reset by powergood reset.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + A0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
Reserved		

2.19 Error Command Register (ERRCMD) — Offset A4h

This register controls the MISA response to various system errors. When a bit in this register is set, an interrupt is generated whenever the corresponding flag is set in the ERRSTS register. The actual generation of the interrupt is globally enabled via the PCI command register.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + A4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
Reserved		

2.20 Top of Upper Usable DRAM Low (B_CR_TOUUD_LO_0_0_0_PCI) — Offset A8h

Defines the top of the upper usable DRAM range and start of the upper MMIO address range. Formerly defined in BMBOUND_HI. This register contains the lower 32 bits of TOUUD.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + A8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW	**Top of Upper Usable DRAM (TOUUD):** TOUUD[38:20] is {TOUUD_HI[6:0],TOUUD_LO[31:20]}. Upper usable DRAM address range ends at the preceding byte. Upper MMIO Range starts at this address and extends up to the maximum system addressable memory range. Bits 38:20 are compared with bits 38:20 of the incoming request address to determine whether the request targets the upper usable DRAM range or upper MMIO range. If Request Address[38:20]>=TOUUD[38:20] and any bit in Request Address[38:32] is set, request is determined to target the upper MMIO range.
19:1	0h RO	**Reserved (RESERVED_1):** Reserved
0	0h RW	**LOCK:** Reserved. Lock is unused by B-Unit. Register overwrites are protected via SAI access control policy registers.

2.21 Top of Upper Usable DRAM High (B_CR_TOUUD_HI_0_0_0_PCI) — Offset ACh

Defines the top of the upper usable DRAM range and start of the upper MMIO address range. Formerly defined in BMBOUND_HI. This register contains the upper 32 bits of TOUUD.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + ACh	1 h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Reserved (RESERVED_0):** Reserved
6:0	1h RW	**Top of Upper Usable DRAM (TOUUD):** TOUUD[38:20] is {TOUUD_HI[6:0],TOUUD_LO[31:20]}. Upper usable DRAM address range ends at the preceding byte. Upper MMIO Range starts at this address and extends up to the maximum system addressable memory range. Bits 38:20 are compared with bits 38:20 of the incoming request address to determine whether the request targets the upper usable DRAM range or upper MMIO range. If Request Address[38:20]>=TOUUD[38:20] and any bit in Request Address[38:32] is set, request is determined to target the upper MMIO range.

2.22 Base of Data Stolen Memory (BDSM_0_0_0_PCI) — Offset B0h

Defines the Base of the Data Stolen Memory. In addition, the GSM range is defined to end at the preceding byte. See BGSM register.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + B0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW	**Base of Data Stolen Memory (BDSM):** This register contains the base address of the Data Stolen Memory. The limit for the Data Stolen Memory is TOLUD-1. This range is not decoded by the system agent, but is a sub-region of BGSM decoded by the Integrated Graphics Device. Incoming Request Address[31:20] is compared against BDSM[31:20] and TOLUD[31:20] to determine if the address falls in the range. The comparison check is as follows: BDSM[31:20]<=Address[31:20] && Address[31:20]<TOLUD[31:20] and Address[38:32]==0. Request SAI is then checked against the allowed SAIs to determine if access is allowed.
19:1	0h RO	**Reserved (RESERVED_0):** Reserved
0	0h RW	**BDSM Register Lock (LOCK):** This lock bit only impacts the Display copy of this register. In the C-Unit all register protection is implemented with SAI policy groups. This bit is maintained in the C-Unit for software observability. Display description: This bit will lock all writeable settings in this register, including itself.

2.23 Base of Graphics Stolen Memory (B_CR_BGSM_0_0_0_PCI) — Offset B4h

Defines the Base of the Graphics Stolen Memory. In addition, the SMM range is defined to end at the preceding byte. See TSEGMB register.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + B4h	7B800000 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	7B8h RW	**Base of Graphics Stolen Memory (BGSM):** This register contains the base address of the Graphics Stolen Memory. The limit for the Graphics Stolen Memory is TOLUD-1. Incoming Request Address[31:20] is compared against BGSM[31:20] and TOLUD[31:20] to determine if the address falls in the range. The comparison check is as follows: BGSM[31:20]<=Address[31:20] && Address[31:20]<TOLUD[31:20] and Address[38:32]==0. Request SAI is then checked against the allowed SAIs to determine if access is allowed.
19:1	0h RO	**Reserved (RESERVED_0):** Reserved
0	0h RW	**LOCK:** Reserved. Lock is unused by B-Unit. Register overwrites are protected via SAI access control policy registers.

2.24 B-Unit Copy of the TSEG Memory Base (B_CR_TSEGMB_0_0_0_PCI) — Offset B8h

B-Unit copy of the TSEGMB. TSEGMB defines the base of the SMM range and the BGSM register defines the limit of the SMM range. Note: In prior SoCs the base register was defined in SMMRRL register and the limit was defined in the SMMRRH register.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + B8h	7B000000 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	7B0h RW	**Base address of TSEG DRAM Memory (TSEGMB):** BIOS determines the base of TSEG memory which must be at or below Graphics Base of GTT Stolen Memory BGSM. SMM range starts at this base and ends at BGSM1. Incoming Request Address [31:20] will be compared with TSEGMB[31:20] and BGSM[31:20] to determine if the request targets the SMM range. The comparison check is as follows: Address[31:20]>=TSEGMB[31:20] && Address[31:20]<BGSM[31:20] and Address[38:32]==0. If the check passes, the request targets the SMM range. If the protection for the range is enabled, then the request SAI is compared against allowed SAIs specified by BSMRRAC and BSMRWAC registers to determine if access is allowed.
19:1	0h RO	**Reserved (RESERVED_0):** Reserved
0	0h RW	**LOCK:** Reserved. Lock is unused by B-Unit. Register overwrites are protected via SAI access control policy registers.

2.25 Top of Lower Usable DRAM (B_CR_TOLUD_0_0_0_PCI) — Offset BCh

This register defines the Top of Lower Usable DRAM range and start of the Lower MMIO Address range. Formerly defined in BMBOUND.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + BCh	80000000 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	800h RW	**Top of Lower Usable DRAM (TOLUD):** Defines the top of lower usable DRAM, which ends at the preceding byte. Lower MMIO Address range starts at this address and continues up to the 4GB Address 0xFFFF_FFFF. Bits 31:20 are compared with incoming request Address[31:20] to determine whether the request targets lower usable DRAM range or the lower MMIO range. If Request Address[31:20]>=TOLUD[31:20] and Request Address[38:32]==0 then the Request Address falls in the Lower MMIO Address range.
19:1	0h RO	**Reserved (RESERVED_0):** Reserved
0	0h RW	**LOCK:** Reserved. Lock is unused by B-Unit. Register overwrites are protected via SAI access control policy registers.

2.26 B-Unit Copy of the MCHBAR (B_CR_MCHBAR_LO_SHADOW_0_0_0_PCI) — Offset D0h

This register contains the lower 32bits of the MCHBAR.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + D0h	FED10001 h

Bit Range	Default & Access	Field Name (ID): Description
31:15	1FDA2h RW	**Base Address of MCHBAR (BASE_ADDR):** Defines the base address of the MCHBAR. MCHBAR[38:15] is {MCHBAR_HI[6:0],MCHBAR_LO[31:15]}. If incoming Request Address[38:15] matches MCHBAR[38:15] then request hits an address in the MCHBAR range.
14:4	0h RO	**Base Address Not Implemented (BASE_ADDR_NOT_IMPLEMENTED):** Hardwired to 0 to indicate size of BAR 32kB.
3:1	0h RO	**Reserved (RESERVED_0):** Reserved
0	1h RW	**MCHBAR Enable (MCHBAREN):** • 0: MCHBAR is disabled and does not claim any memory • 1: MCHBAR memory mapped accesses are claimed and decoded appropriately

2.27 B-Unit Copy of the MCHBAR (B_CR_MCHBAR_HI_SHADOW_0_0_0_PCI) — Offset D4h

This register contains the upper 32 bits of the MCHBAR.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + D4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Reserved (RESERVED_0):** Reserved
6:0	0h RW	**Base Address of MCHBAR (BASE_ADDR):** Defines the base address of the MCHBAR. If incoming Request Address[38:15] matches MCHBAR[38:15], then request hits an address in the MCHBAR range.

2.28 Message Data Register (MCRX) — Offset D8h

This register provides the extensions to complete the message fields over SBI. Its contents are used to fill in the upper address or offset bits of the SBI request and are cleared upon a write to the MCR

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + D8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
Reserved		

2.29 Scratchpad (SKPD_0_0_0_PCI) — Offset DCh

This register holds 32 writable bits with no functionality behind them. It is for the convenience of BIOS and graphics drivers.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + DCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Scratchpad (SKPD):** 1 DWORD of data storage.

2.30 Capability ID0 Capability Control (CAPID0_CAPCTRL0_0_0_0_PCI) — Offset E0h

Control bits in this register describe the attributes of CAPID0_A and CAPID0_B capability registers.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + E0h	10C0009 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RESERVED_0):** Reserved
27:24	1h RO	**Capability ID Version (CAPID_VER):** This field has the value 0001b to identify the first revision of the CAPID register definition.
23:16	Ch RO	**Capability ID0 Structure Length (CAPIDLEN):** This field has the value 0Ch to indicate the structure length 12 bytes. This is the total size of this CAPCTRL and the CAPID0_A and CAPID0_B registers in the following bytes.
15:8	0h RO	**Next Capability Register Pointer (NEXT_CAP):** This field is hardwired to 00h, indicating the end of the capabilities linked list.
7:0	9h RO	**Capability ID (CAP_ID):** This field has the value 1001b to identify the CAP_ID assigned by the PCI SIG for vendor dependent capability pointers.

2.31 Capability ID0 A (B_CR_CAPID0_A_0_0_0_PCI) — Offset E4h

Control of bits in this register is only required for SKU differentiation.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + E4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**Spare 31-24 (SPARE31_24):** Reserved for future capabilities.
23	0h RW	**VTd Disable (VTDD):** • 0: Enable VTd • 1: Disable VTd
22	0h RW	**Fuse Spare 22 (FUSE_SPARE22):** Fuse backed spare.
21	0h RW	**Fuse Spare 21 (FUSE_SPARE21):** Fuse backed spare.
20	0h RW	**Fuse Spare 20 (FUSE_SPARE20):** Fuse backed spare.
19	0h RW	**Fuse Spare 19 (FUSE_SPARE19):** Fuse backed spare.
18	0h RW	**Fuse Spare 18 (FUSE_SPARE18):** Fuse backed spare.
17:16	0h RO	**Reserved (RSVD):** Reserved
15	0h RW	**Camarillo DPTF Disable (CDD):** • 0: DPTF Camarillo associated memory spaces are accessible. • 1: DPTF Camarillo associated memory and IO spaces are disabled. DEVEN_0_0_0_PCI field for DPTF can not be set.
14	0h RW	**Fuse Spare 14 (FUSE_SPARE14):** Fuse backed spare.
13	0h RW	**Fuse Spare 13 (FUSE_SPARE13):** Fuse backed spare.
12	0h RW	**Fuse Spare 12 (FUSE_SPARE12):** Fuse backed spare.
11	0h RW	**Internal Graphics Disable (IGD):** • 0: There is a graphics engine within this CPU. Internal Graphics Device 2 is enabled, and all of its memory and I/O spaces are accessible. Configuration cycles to Device 2 will be completed within the CPU. All nonSMM memory and IO accesses to VGA will be handled based on Memory and IO enables of Device 2, IO registers within Device 2, and VGA Enable of the PCI to PCI bridge control register in Devices 1 and 6, if PCI Express GFX attach is supported. A selected amount of Graphics Memory space is preallocated from the main memory, based on Graphics Mode Select GMS in the GGC Register. Graphics Memory is preallocated above TSEG Memory. • 1: There is no graphics engine within this CPU. Internal Graphics Device 2 and all of its memory and I/O functions are disabled. Configuration cycles targeted to Device 2 will be passed on to DMI. In addition, all clocks to internal graphics logic are turned off. All nonSMM memory and IO accesses to VGA will be handled based on VGA Enable of the PCI to PCI bridge control register in Devices 1 and 6. DEVEN [4:3] Device 0 offset 54h have no meaning. Device 2 Functions 0 and 1 are disabled and hidden.
10	0h RO	**Device ID Override Enable (DIDOE):** Controls if there is an override of Dev2 GFX device ID. Hardwired to 1b0. • 0: Disable ability to override DID -- For production • 1: Enable ability to override DID -- For debug and samples only

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RO	**Control Device ID Value (CDID):** Controls the value of Dev2 GFX device ID. Hardwired to 2b00. Identifier assigned to the core/primary PCI device.The corresponding two bit capability ID programming is: • 00: Desktop • 01: Mobile • 10: Server • 11: Marketing Spare
7:0	0h RW	**Spare 7-0 (SPARE7_0):** Reserved for future capabilities.

2.32 Capability ID0 B (B_CR_CAPID0_B_0_0_0_PCI) — Offset E8h

Control of bits in this register is only required for SKU differentiation.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + E8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Imaging Unit Memory/IO Disable (IMGU_DIS):** • 0: Imaging Unit associated memory spaces are accessible. • 1: Imaging Unit associated memory and IO spaces are disabled. DEVEN_0_0_0_PCI field for Iunit can not be set.
30	0h RW	**Fuse Spare 30 (FUSE_SPARE30):** Fuse backed spare.
29	0h RW	**Fuse Spare 29 (FUSE_SPARE29):** Fuse backed spares potentially to be used for PKGTYP encoding.
28	0h RW	**Fuse Spare 28 (FUSE_SPARE28):** Fuse backed spares potentially to be used for PKGTYP encoding.
27	0h RW	**Fuse Spare 27 (FUSE_SPARE27):** Fuse backed spares potentially to be used for PKGTYP encoding.
26	0h RW	**Fuse Spare 26 (FUSE_SPARE26):** Fuse backed spares potentially to be used for PKGTYP encoding.
25	0h RW	**Fuse Spare 25 (FUSE_SPARE25):** Fuse backed spares potentially to be used for PKGTYP encoding.
24	0h RW	**SVM Disable (SVMDIS):** • 0: Enable SVM mode • 1: Disable SVM mode
23:0	0h RW	**Spare 23-0 (SPARE23_0):** Reserved for future capabilities.

2.33 Design and Engineering Backup Register 0 (DEBUP0_0_0_0_PCI) — Offset F4h

Reserved for future spare usage.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + F4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Spare bits[31:0] (SPARE_RW):** Spare RW bits for future usage.

2.34 Design and Engineering Backup Register 1 (DEBUP1_0_0_0_PCI) — Offset FCh

Reserved for future spare usage.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:0] + FCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Spare bits[31:0] (SPARE_RW):** Spare RW bits for future usage.

§ §

3 Host Memory Mapped Configuration Space (MCHBAR) Registers

This chapter documents the MCHBAR registers. Base address of these registers are defined in the MCHBAR_0_0_0_PCI register in Bus: 0, Device: 0, Function: 0.

NOTE: These registers apply to all processors.

3.1 DRAM Rank Population 0 (D_CR_DRP0) — Offset 1000h

Rank configuration register.

This register is start of dunit 8, LPDDR4 Channel 1.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1000h	10000000 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RW	**DRAM Device Per Rank (DRAMDEVICE_PR):** Specifies the number of DRAM devices that are ganged together to form a single rank. • 00: 1 DRAM device in each rank. • 01: 2 DRAM devices in each rank. • 10: 4 DRAM devices in each rank. • 11: 8 DRAM devices in each rank. Note: The actual number of devices is one more than the value programmed when ECC is enabled.
29:28	1h RW	**Address Decode (ADDRDEC):** Specifies the address mapping to be used: • 00: 1KB (A). • 01: 2KB (B). • 10: 4KB (C). • 11: Reserved.
27:25	0h RW	**Burst Length Mode (BLMODE):** • 000: Fixed BL8. • 001: Onthefly BL8. • 010: Fixed BL16. • 011: Onthefly BL16. • 100: Fixed BL32. • 101: Onthefly BL32. • 110-111: Reserved.

Bit Range	Default & Access	Field Name (ID): Description
24:22	0h RW	**DRAM Type (DRAMTYPE):** • 000: Reserved. • 001: Reserved. • 010: LPDDR4. • 011: Reserved. • 100: DDR4. • 101-111: Reserved. Note: The D-Unit should only use this field if allowed by fuse.
21	0h RW	**ECC Enable (ECCEN):** • 0: ECC is disabled. • 1: ECC is enabled. This bit determines if the D-Unit treats the PMI BE_ECC bits as ECC bits or Byte Enables. The D-Unit should not allow this bit to be set if ECC is disabled by fuse. This should only be used in configurations that support ECC.
20:19	0h RW	**CA Swizzle Type (CASWIZZLE):** • 00: uniDIMM/SODIMM/UDIMM. • 01: BGA. • 10: NA. • 11: uniDIMM/SODIMM/UDIMM with Rank 1 mirrored (DDR4 Only).
18:17	0h RO	**Reserved (RSVD18_17):** Reserved
16	0h RW	**Fine Bank Group Interleaving Enable (FBGINTEN):** When enabled, D-unit will use a lower order address bit in the bank hashing function (DDR4 Only). • 0: Bank Group Interleave disabled. • 1: Bank Group Interleave enabled. Note: BAHEN must be set for this bit to take effect.
15	0h RW	**Bank Address Hashing Enable (BAHEN):** • 0: Bank Address Hashing disabled. • 1: Bank Address Hashing enabled.
14	0h RW	**Rank Select Interleave Enable (RSIEN):** • 0: Rank Select Interleaving disabled. • 1: Rank Select Interleaving enabled.
13:9	0h RO	**Reserved (RSVD13_9):** Reserved
8:6	0h RW	**DRAM Device Density (DDEN):** Density of the DRAM devices populated on Ranks 0 and 1. • 000: 4 Gb. • 001: 6 Gb. • 010: 8 Gb. • 011: 12 Gb. • 100: 16 Gb. • 101-111: Reserved. Note: For LPDDR4 this value is the die density.
5:4	0h RW	**DRAM Device Data Width (DWID):** Data width of the DRAM device populated on Ranks 0 and 1. • 00: x8. • 01: x16. • 10: x32. • 11: x64.
3	0h RO	**Reserved (RSVD3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Dual Data Mode Enable (DDMEN):** • 0: PMI Dual Data Mode is disabled in D-Unit. • 1: PMI Dual Data Mode is enabled. Note: Dual Data Mode must be enable for DDR3L/DDR4 configurations.
1	0h RW	**Rank Enable 1 (RKEN1):** Enable Rank 1: Must be set to 1 to enable use of this rank.
0	0h RW	**Rank Enable 0 (RKEN0):** Enable Rank 0: Must be set to 1 to enable use of this rank. Note: Setting this bit to 0 is not a functional mode.

3.2 DRAM Timing Register 9A (D_CR_DTR9A) — Offset 1004h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1004h	14A546 h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RO	**Reserved (RSVD31_23):** Reserved
22:18	5h RW	**Write to Write DQ Delay Same Bank Group Same Rank (TWWSR_L):** Specifies the delay from a DRAM Write to another Write command within the same bank group of the same rank (in DRAM clocks). DDR4 Equation: tWWSR_L = tCCD_L.
17:13	5h RW	**Read to Read DQ Delay Same Bank Group Same Rank (TRRSR_L):** Specifies the delay from a DRAM Read to another Read command within the same bank group of the same rank (in DRAM clocks). DDR4 Equation: tRRSR_L = tCCD_L.
12:6	15h RW	**Write to Read DQ Delay Same Bank Group Same Rank (TWRSR_L):** Specifies the delay from a DRAM Read to Write command within the same bank group of the same rank (in DRAM clocks). • DDR4 Equation: tWRSR_L = CWL + tDQSSmax + BL/2 + tWPST + tWTR_L.
5:0	6h RW	**Row Activation to Row Activation to Same Bank Group Delay [tRRD_L] (TRRD_L):** Specifies the minimum delay (in DRAM clocks) between two DRAM Activate commands to different banks within the same bank group of a rank. (DDR4 Only)

3.3 DRAM Timing Register 0A (D_CR_DTR0A) — Offset 1008h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1008h	820702CB h

Bit Range	Default & Access	Field Name (ID): Description
31:27	10h RW	**Valid Clocks Before CKE High [tCKCKEH/tCSCKEH/tCKSRX] (TCKCKEH):** Number of valid clocks before CKE high (in DRAM clocks). • LPDDR4: The value in this register covers both tCKCKEH and tCSCKEH. • DDR3L/DDR4/LPDDR3/WIO2: The value covers tCKSRX which is defined as the number of valid DRAM clocks that have to toggle before the issuing of the Self Refresh Exit SRX. This value is also used if the clock frequency is changed or the clock is stopped or tristated during Power Down i.e. the number valid DRAM clocks that have to toggle before the issuing of the Power Down Exit PDX command. TCKCKEH can be used to compensate for clock stabilization delays in the motherboard. Note: D-unit hardware enforces minimum of two SPID clock before CKEH, any value in this register is the additional time.
26:22	8h RW	**Exit Self-Refresh to Valid Commands Requiring a Locked DLL Delay [tXSDLL] (TXSDLL):** D-Unit waits max(tXSR+tZQCL/tZQCS, tXSDLL) before allowing traffic to DRAM (in 32 x DRAM Clocks). LPDDR3/LPDDR4/WIO2: tXSDLL = 0. DDR3L/DDR4: tXSDLL = tDLLK. Note: In the equation above, tZQCL/tZQCS = 0 if no ZQ is performed on SR exit.
21:12	70h RW	**Exit Self-Refresh to Valid Command Delay [tXS/tXSR] (TXSR):** DDR3L/DDR4: tXS - Delay between Self Refresh Exit SRX to any DRAM Command not requiring DLL Lock. LPDDR/WIO2: tXSR - Delay between Self Refresh Exit SRX to any DRAM Command. (in DRAM clocks).
11:6	Bh RW	**Activate RAS to CAS Command Delay [tRCD] (TRCD):** Specifies the delay between a DRAM Activate command and a DRAM Read or Write command to the same bank (in DRAM clocks). Note: Derating adds 1.875ns to this timing.
5:0	Bh RW	**Precharge to Activate Command Delay of a Single Bank [tRPpb] (TRPPB):** Specifies the delay between a DRAM Precharge command and a DRAM Activate command to the same bank (in DRAM Clocks). Note: this CR should be constrained to a minimum of 4 in LPDDR3/DDR3L/WIO2 and minimum of 8 in LPDDR4/DDR4. Note: Derating adds 1.875ns to this timing.

3.4 DRAM Timing Register 1A (D_CR_DTR1A) — Offset 100Ch

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 100Ch	30481218 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	6h RW	**Exit Power Down to Next Command Delay [tXP] (TXP):** Specifies the delay from the DRAM Power Down Exit (PDX) command to any valid command (in DRAM clocks). Note: The value in this field must be programmed to tXPDLL when Slow Exit Mode Power-down is enabled for DDR3L.
26	0h RO	**Reserved (RSVD26):** Reserved
25:14	120h RW	**ZQ (long) Calibration Time [tZQCL/tZQCAL] (TZQCL):** • LPDDR3/DDR3L/DDR4: tZQCL/tZQoper - Specifies the delay between the DRAM ZQ Calibration Long (ZQCL) command and any DRAM command during normal operation. • LPDDR4: tZQCAL - ZQ Calibration time (in DRAM clocks). Note: This field defines the ZQ Calibration Long delay during normal operation. It is not the same as tZQinit which uses the same ZQCL command but the delay is longer. tZQinit applies only during poweron initialization of the DRAM devices and tZQoper applies during normal operation. BIOS executes the DRAM initialization sequence so it has to ensure tZQinit is met and not the D-Unit.

Bit Range	Default & Access	Field Name (ID): Description
13:6	48h RW	**ZQ Short Calibration Time [tZQCS] (TZQCS):** ZQCS to any DRAM Command Delay: Specifies the delay between the DRAM ZQ Calibration Short (ZQCS) command and any DRAM command (in DRAM clocks). DDR3L/DDR4/LPDDR3 only. LPDDR4 does not support ZQCS command
5:0	18h RW	**ZQ Latch Time [tZQLAT] (TZQLAT):** Specifies the delay between the DRAM ZQ Calibration Latch command and any DRAM command (in DRAM clocks). LPDDR4 only.

3.5 DRAM Timing Register 2A (D_CR_DTR2A) — Offset 1010h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1010h	46080C30 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	118h RW	**All Bank Refresh Cycle Time [tRFCab] (NRFCAB):** Specifies the delay between the REFab command to the next valid command. (in DRAM clocks)
21	0h RO	**Reserved (RSVD21):** Reserved
20:17	4h RW	**CKE Minimum Pulse Width [tCKE] (TCKE):** Specifies the minimum time from CKEL to CKEH (in DRAM clocks).
16	0h RO	**Reserved (RSVD16):** Reserved
15:0	C30h RW	**Refresh Interval Time [tREFI] (NREFI):** Specifies the average time between refresh commands. JEDEC Base Refresh Interval time (in DRAM clocks). Note: D-Unit will ignore the 2 LSBs of this field.

3.6 DRAM Timing Register 3A (D_CR_DTR3A) — Offset 1014h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1014h	3002EA28 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	6h RW	**Read to Precharge Delay [tRDPRE] (TRTP):** Specifies the minimum delay between the DRAM Read and Precharge commands to the same bank (in DRAM clocks). • LPDDR3 Equation: = NA. • LPDDR4 Equation: = BL/2 + Max (8, tRTP) - 8. • WIO2 Equation: = NA. • DDR3L/DDR4 Equation: = tRTP.

Bit Range	Default & Access	Field Name (ID): Description
26:20	0h RW	**CAS to CAS Command Delay Adder (TCCD_INC):** Specifies the number of clocks to be added to turnaround times (for Stretch Mode). It increases delay between Read to Read or Read to Write commands by the amount programmed in this field (in 4 x DRAM clocks).
19:13	17h RW	**Write to Precharge Command Delay [tWRPRE] (TWTP):** Specifies the minimum delay between the DRAM Write command and the Precharge command to the same bank (in DRAM clocks). • LPDDR3/LPDDR4/WIO2 Equation: tWTP = BL/2 + WL + tWR + 1. • DDR3L/DDR4 Equation: tWTP = BL/2 + CWL + tWR.
12:11	1h RW	**DRAM Command Valid Duration (TCMD):** Specifies the number of DRAM clocks a command is held valid on the DRAM Address and Control buses. 1N is the DDR3 basic requirement. 2N is the extended mode for board signal integrity. • 0h: Reserved. • 1h: 1 DRAM Clock (1N). • 2h: 2 DRAM Clocks (2N). • 3h: Reserved. Note: DDR3L/DDR4 only. tCMD must be set to 1N for LPDDR3/LPDDR4/WIO2.
10:6	8h RW	**Write Latency [WL/CWL] (TCWL):** The delay between the internal write command and the availability of the first word of DRAM input data (in DRAM clocks).
5:0	28h RW	**Write CAS to Masked Write CAS Delay Same Bank (TWMWSB):** Specifies the minimum delay between DRAM Write command to Masked Write command to same bank (in DRAM clocks). • LPDDR4 Equation: tWMWSB = tCCDMW (BL16) or tCCDMW + 8 (BL32). • WIO2 Equation: NA. Note: Masked Write operation in LPDDR4 is always BL16 and in WIO2 is always BL4. D-Unit applies this timing for same rank as well as same bank.

3.7 DRAM Timing Register 4A (D_CR_DTR4A) — Offset 1018h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1018h	30209149 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	30h RW	**Four Bank Activate Window [tFAW] (TFAW):** A rolling timeframe in which a maximum of four Activate commands can be issued to the same rank. This is to limit the peak current draw from the DRAM devices (in DRAM clocks).
23:18	8h RW	**Write to Read DQ Delay Different Ranks (TWRDR):** Specifies the delay from the start of a Write data burst of one rank to the start of a Read data burst of a different rank (in DRAM clocks). • LPDDR3 Equation: tWRDR = WL + tDQSSmax + BL/2 - (RL + tDQSCKmin - tRPRE). • LPDDR4 Equation: tWRDR = WL - RL + BL/2 + 4 - tDQSCKmin. • DDR3L/DDR4 Equation: tWRDR = CWL + tDQSSmax + BL/2 - (CL + tDQSCKmin - tRPRE). Note: For LPDDR3/4 using ODT, this latency may need to be increased by tODToffadj.

Bit Range	Default & Access	Field Name (ID): Description
17:12	9h RW	**Read to Write DQ Delay Different Ranks (TRWDR):** Specifies the delay from the start of a Read data burst of one rank to the Start of a Write data burst of a different rank (in DRAM clocks). • LPDDR3/LPDDR4 Equation: tRWDR = RL + tDQSCKmax + BL/2 - (WL - tWPRE). • DDR3L/DDR4 Equation: tRWDR = CL + tDQSCKmax + BL/2 - (CWL - tWPRE). Note: For LPDDR3/4 using ODT, this latency may need to be adjusted by tODTon. Note: For DDR3L/DDR4 using ODT, this latency may need to be increased by one clock.
11:6	5h RW	**Write to Write DQ Delay Different Ranks (TWWDR):** Specifies the delay from the start of a Write data burst of one rank to the start of a Write data burst of a different rank (in DRAM clocks). • LPDDR3/DDR3L/DDR4 Equation: tWWDR = BL/2 + tDQSSmax - tDQSSmin + tWPRE. • LPDDR4 Equation: tWWDR = BL/2 + 4 - tDQSSmin. Note: For LPDDR3/4 using ODT, this latency may need to be increased by tODToffadj.
5:0	9h RW	**Read to Read DQ Delay Different Ranks (TRRDR):** Specifies the delay from the start of a Read data burst of one rank to the Start of a Read data burst of a different rank (in DRAM clocks). • Equation: tRRDR = BL/2 + tDQSCKmax - tDQSCKmin + tRPRE.

3.8 DRAM Timing Register 5A (D_CR_DTR5A) — Offset 101Ch

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 101Ch	304200C2 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	6h RW	**Row Activation to Row Activation Delay [tRRD] (TRRD):** Specifies the minimum delay in DRAM clocks between two DRAM Activate commands to the same rank but different banks (tRC is the minimum delay between activations of the same bank). Note: Derating adds 1.875ns to this timing.
26	0h RO	**Reserved (RSVD26):** Reserved
25:23	0h RW	**Derate Increment (TDERATE_INC):** Specifies the additional delay that is added to DRAM timing when indicated by MR4 status. (in DRAM clocks) LPDDR3/LPDDR4/WIO2: Value is 1.875ns. Note: The value in this register is only added to these timing parameters: tRCD, tRAS, tRP and tRRD.
22:18	10h RW	**Write to Write DQ Delay Same Rank (TWWSR):** Specifies the delay from a DRAM Write to another Write command of the same rank (in DRAM clocks). Equation: tWWSR = tCCD.
17:13	10h RW	**Read to Read DQ Delay Same Rank (TRRSR):** Specifies the delay from a DRAM Read to another Read command of the same rank (in DRAM clocks). Equation: tRRSR = tCCD.
12:6	3h RW	**Write to Read DQ Delay Same Rank (TWRSR):** Specifies the delay from a DRAM Read to Write command of the same rank (in DRAM clocks). • LPDDR3/LPDDR4/WIO2 Equation: tWRSR = WL + tDQSSmax + BL/2 + tWTR. • DDR3L Equation: tWRSR = CWL + tDQSSmax + BL/2 + tWTR. • DDR4 Equation: tWRSR = CWL + tDQSSmax + BL/2 + tWTR_S.

Bit Range	Default & Access	Field Name (ID): Description
5:0	2h RW	**Read to Write DQ Delay Same Rank (TRWSR):** Specifies the delay from a DRAM Read to a Write command of the same rank (in DRAM clocks). • LPDDR3/LPDDR4/WIO2 Equation: tRWSR = RL + tDQSCKmax + BL/2 - WL + tWPRE. • DDR3L/DDR4 Equation: tRWSR = CL + tDQSCKmax + BL/2 - CWL + tWPRE. Note: For LPDDR3/4 using ODT, this latency may need to be increased by tODToffadj. Note: For DDR3L/DDR4 using ODT, this latency may need to be increased by one clock.

3.9 DRAM Timing Register 6A (D_CR_DTR6A) — Offset 1020h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1020h	20100000 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	20h RW	**Opportunistic Refresh Idle Timer (OREFDLY):** Rank idle period that defines an opportunity for refresh (in DRAM clocks).
23:19	2h RW	**Valid Clocks After CKE Low [tCKELCK/tCKELCS/tCPDED/tCKSRE] (TCKCKEL):** Specifies the amount of time that DRAM clocks need to toggle after CKE goes low (in DRAM Clocks). • For WIO2/LPDDR3, this covers tCPDED. • For LPDDR4, this covers both tCKELCK and tCKELCS. • For DDR3L/DDR4, this is tCKSRE. Note: D-Unit hardware enforces minimum of one SPID clocks after CKEL, any value in this register is the additional time.
18:15	0h RW	**Reserved (RSVD18_15):** Reserved
14:8	0h RW	**Mode Register Read to Any Command Delay (TPSTMRRBLK):** Specifies the quiet time after issuing MRR command (in DRAM Clocks). This is the channel block time for the MR19 command issued as part of LPDDR4 DQS Retraining flow. For all other MR commands this is the rank block time. Note: D-Unit treats MRR as a read and always applies relevant turnaround times, any value programmed in this CR must be greater than those turnaround times for D-Unit to enforce any additional time from MRR to the next read/write.
7	0h RO	**Reserved (RSVD7):** Reserved
6:0	0h RW	**Any Command to Mode Register Read/Write Delay (TPREMRBLK):** Specifies the channel quiet time before issuing MRR/MRW command. (in DRAM clocks). • LPDDR3 Equation: = tXP + tMRRI - 4. • LPDDR4 Equation: = tXP + tMRRI - 8. • DDR3L Equation: = CWL + BL/2 + 1 - 8 • DDR4 Equation: = CWL + BL/2 + 1 - 16. Note: D-Unit treats MRR as a read and MRW/MRS as a write and always applies relevant turnaround times, any value programmed in this CR must be greater than those turnaround times for D-Unit to enforce any additional time from previous read/writes.

3.10 DRAM Timing Register 7A (D_CR_DTR7A) — Offset 1024h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1024h	D060C06 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	3h RW	**All Bank Precharge to Activate Command Delay [tRPab] (TRPAB):** Specifies the delay between a DRAM Precharge All Bank command and a DRAM Activate command (in DRAM Clocks). Note: This CR should be constrained to a minimum of 4 in LP3 and minimum of 8 in LP4. Note: Derating adds 1.875ns to this timing. • For LPDDR, tRPpb = tRP, tRPab = tRP + 3ns. • For DDR3L, DDR4 and WIO2 8ch tRPpb = tRPab = tRP.
25:23	2h RW	**Mode Register Write to any Command Delay [tMRD/tMRW] (TPSTMRWBLK):** Specifies the quiet time after issuing MRW command (in 8 x DRAM clocks). Note: This time covers for both tMRD and tMRW.
22:16	6h RW	**Write Command to Power Down Delay [tWRPDEN] (TWRPDEN):** Specifies the minimum time between a write command to PowerDown command (in DRAM clocks). Must be at least equal to tWR + tCCD + tWL + 2.
15:9	6h RW	**Read Command to Power Down Delay [tRDPDEN] (TRDPDEN):** Specifies the minimum time between a read command to PowerDown command (in DRAM clocks). Must be at least equal to CL/RL + tDQSCKmax + tCCD + 1.
8:7	0h RO	**Reserved (RSVD8_7):** Reserved
6:0	6h RW	**Row Activation Period [tRAS] (TRAS):** Specifies the minimum delay between the DRAM Activate and Precharge commands to the same bank (in DRAM clocks). Note: Derating adds 1.875ns to this timing.

3.11 DRAM Timing Register 8A (D_CR_DTR8A) — Offset 1028h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1028h	CC50A18 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	3h RW	**Minimum Self-Refresh Time [tSR/tCKESR] (TCKESR):** Specifies the minimum time that DRAM should remain in SR (in DRAM clocks).
25:21	6h RW	**Minimum Low Power Mode Residency (LPMDRES):** Specifies the minimum time that PHY should remain in LPMode (in DRAM clocks).
20:15	Ah RW	**Low Power Mode Exit to Clock Enable Delay (LPMDTOCKEDLY):** Specifies the minimum time between the LP Mode exit to the CK stop/tristate deassertion and powerdown exit (in DRAM clocks). Note: Must be equal to t_idle_latency published in the DDRIO PHY and less than 0x3C.
14:8	Ah RW	**Clock Stop to Low Power Mode Delay (CKETOLPMDDLY):** Specifies the time between CK stop/tristate to the Low Power Mode entry. This timing parameter is used to delay Low Power Mode entry (in DRAM clocks). Note: Must be at least equal to t_idle_length parameter published in the DDRIO PHY and less than 0x7C.
7:0	18h RW	**Power Down Idle Timer (PWDDLY):** This is a non-JEDEC timing parameter used to delay powerdown entry (in DRAM clocks).

3.12 D-Unit ODT Control Register A (D_CR_DOCRA) — Offset 102Ch

Specifies the parameters to control DRAM ODT.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 102Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29	0h RW	**Rank 1 Read ODT Control (R1RDODTCTL):** Specifies the behavior of ODT signals when a Read command is issued to Rank 1. 0 - Read ODT is disabled for Rank 1 1 - Assert ODT to for Rank 0 (non-targeted Rank) Note: This register should be set to 0 for LPDDR3 devices
28	0h RW	**Rank 0 Read ODT Control (R0RDODTCTL):** Specifies the behavior of ODT signals when a Read command is issued to Rank 0. 0 - Read ODT is disabled for Rank 0 1 - Assert ODT to for Rank 1 (non-targeted Rank) Note: This register is reserved for LPDDR3 devices
27:26	0h RW	**Rank 1 Write ODT Control (R1WRODTCTL):** Specifies the behavior of ODT signals when a Write command is issued to Rank 1. 00 - Write ODT is disabled 01 - Assert ODT to Rank 0 (non-targeted Rank) 10 - Assert ODT to Rank 1 (targeted Rank) 11 - Assert ODT to Rank 0 and Rank 1 Note: 10 and 11 are reserved values for LPDDR3
25:24	0h RW	**Rank 0 Write ODT Control (R0WRODTCTL):** Specifies the behavior of ODT signals when a Write command is issued to Rank 0. 00 - Write ODT is disabled 01 - Assert ODT to Rank 0 (targeted Rank) 10 - Assert ODT to Rank 1 (non-targeted Rank) 11 - Assert ODT to Rank 0 and Rank 1 Note: 10 and 11 are reserved values for LPDDR3
23:18	0h RO	**Reserved (RSVD23_18):** Reserved
17:14	0h RW	**Read ODT assertion to de-assertion delay (DDR3L/DDR4) (RDODTSTOP):** Specifies Read ODT assertion to ODT de-assert delay (in DRAM clocks). DDR3L Equation: RDODTSTOP = 6. DDR4 Equation: RDODTSTOP = 5 + tRPRE. Note: Add 1 if DOCRx.RDODTSTART = CL - CWL in 2N mode.
13	0h RO	**Reserved (RSVD13):** Reserved
12:9	0h RW	**Read command to ODT assertion delay (DDR3L/DDR4) (RDODTSTART):** Specifies Read ODT assertion delay after Read Command (in DRAM clocks). DDR3L/DDR4 Equation: RDODTSTART = CL - CWL + tWPRE - tRPRE. Note: In DDR3L/DDR4 2N mode add 1 to enable termination at the start of read data burst.
8:5	0h RW	**Write ODT Assertion to De-assertion Delay (WRODTSTOP):** Specifies number of clocks after ODT assertion that D-Unit deasserts ODT signal (in DRAM clocks). LPDDR3 Equation: WRODTSTOP = RU(tODTon(max)/tCK) + RU((tDQSSmax+tWPST)/tCK) + BL/2 - RD(tODToffmin/tCK) DDR3L Equation: WRODTSTOP = 6. DDR4 Equation: WRODTSTOP = 5 + tWPRE. Note: Add 1 if DOCRx.WRODTSTART = 0 in 2N mode.
4	0h RO	**Reserved (RSVD4):** Reserved
3:0	0h RW	**Write command to ODT assertion delay (WRODTSTART):** Specifies number of clocks after Write command that D-Unit asserts ODT signal (in DRAM clocks). LPDDR3 Equation: WRODTSTART = WL - RU(tODTon(max)/tCK) DDR3L/DDR4 Equation: WRODTSTART = 0 Note: In DDR3L/DDR4 2N mode the value can be set to 0 to assert ODT one DRAM clock earlier than the Write Command (WR) or set to 1 to assert at the same clock as command (CS assertion).

3.13 D-Unit Power Management Control 0 (D_CR_DPMC0) — Offset 1030h

Specifies the parameters to control D-Unit power management features.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1030h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**Reserved (RSVD31_29):** Reserved
28:24	0h RW	**SUSPEND/SUSPENDP Power Management Message Opcode (SUSPMOP):** DDRIO PHY Power Mode Opcode: After the D-Unit has placed the DRAM devices in Self Refresh/PASR mode as the result of a SUSPEND/SUSPENDP message, it sends this 5-bit value to the DDRIO PHY to tell it which power saving mode it should enter. Changing this register value while in SUSPEND will have no effect. Note: This opcode cannot be a PM state where it disables PHY PLLs i.e PM7 in LPDDR PHY.
23	0h RO	**Reserved (RSVD23):** Reserved
22	0h RW	**PM Message Wait for Clock Gate Enable (SRPMCLKW):** Specifies when it is safe to send PM message to the PHY. When enabled, D-Unit waits for SPID Clock to deassert before sending a PM message on SR entry. • 0: D-Unit will not wait for SPID_clk to deassert before sending the PM message to PHY. • 1: D-Unit will wait for SPID_clk to deassert before sending PM message to the PHY. Note: The value must be 1 when DYNPMOP = 7h.
21:17	0h RW	**Dynamic Self-Refresh Power Management Message Opcode (DYNPMOP):** DDRIO PHY Power Mode Opcode: After the D-Unit has placed the DRAM devices in Self Refresh mode as the result of a Dynamic Self-Refresh, it sends this 5bit value to the DDRIO PHY to tell it which power saving mode it should enter. Changing this register value while in self-refresh will only change the PM state for the next entry in DynSR.
16	0h RW	**Dynamic Self-Refresh Enable (DYNSREN):** When set to 1, the D-Unit will automatically control DRAM Self Refresh entry and exit based on interface state and requests in pending queues. When there is no pending request in the queues and PMI is idle, then the D-Unit will place the DRAM devices in Self Refresh mode. The DRAM devices will be brought out of Self-Refresh when idle conditions don't hold.
15:0	0h RW	**Self-Refresh Entry Delay (SREDLY):** Specifies the minimum time the D-Unit will wait before it enters Dynamic Self-Refresh mode when idle (in 16x DRAM Clocks). Note: The value in this field needs to be minimum of 4 in functional mode and minimum of 50 in PSMI mode.

3.14 D-Unit Power Management Control 1 (D_CR_DPMC1) — Offset 1034h

Specifies the parameters to control D-Unit power management features.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1034h	10000028 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29	0h RW	**D-Unit Repeaters Clock Gate Disable (RPTCLKGTDIS):** Setting this bit to 0 allows majority of the repeaters between D-Unit and PHY to clock gate when there is no activity in order to save power. • 0 - Enable Repeaters clock gating. • 1 - Disable Repeaters clock gating. Note: This is a de-feature bit and should be set to 0 for normal operation. Note: The value should only change after DRAM Timing Registers (DTR) are programmed.
28	1h RW	**IOSF-SB End Point Clock Gate Disable (SBEPCLKGTDIS):** Setting this bit to 0 enables the clock gating of IOSF-SB End Points in D-Unit and CPGC when there is no IOSF-SB activity in order to save power. • 0 - Enable IOSF-SB EP clock gating. • 1 - Disable IOSF-SB clock gating. Note: This is a de-feature bit and should be set to 0 for normal operation. Note: The value should only change after DRAM Timing Registers (DTR) are programmed.
27	0h RW	**Local Clock Gate Disable (CLKGTDIS):** Setting this bit to 0 allows the majority of the D-Unit clocks to be gated off when there is no activity in order to save power. When set to 1, D-Unit clockgating is disabled. • 0: Enable. • 1: Disable. Note: This is a de-feature bit and should be set to 0 for normal operation. Note: The value should only change after DRAM Timing Registers (DTR) are programmed.
26	0h RW	**Chip Select Tristate Enable (CSTRIST):** • 0: The DRAM CS pins associated with the enabled ranks are never tristated. • 1: The DRAM CS pins are tristated when DRAM clock is stopped or tristated. Note: CS is not tristated when global tristate flow is disabled (DCBR.TRISTDIS = 1).
25:24	0h RW	**Command/Address Tristate (CMDTRIST):** • 00: The DRAM CA pins are never tristated. • 01: The DRAM CA pins are only tristated when all enabled CKE pins are low. • 10: The DRAM CA pins are tristated when not driving a valid command. • 11: Reserved
23:16	0h RW	**Partial Array Self-Refresh Segment Mask (PASR):** This is the Segment Mask used for the MRW to enable PASR during SUSPENDP (Partial Array Self Refresh entry).
15:8	0h RW	**Page Close Timeout Period (PCLSTO):** Specifies the time from the last access of a DRAM page until that page is scheduled to close by sending a Precharge command to DRAM (in 16 x DRAM clocks).
7	0h RW	**Page Close Timeout Disable (PCLSTODIS):** When disabled, D-Unit will not close the DRAM page when idle. • 0: Enable page close timer. • 1: Disable page close timer (Used during DRAM init and DDRIO training).
6	0h RO	**Reserved (RSVD6):** Reserved
5	1h RW	**ODT Tristate Enable (ODTTRIST):** • 0: The DRAM ODT pins associated with the enabled ranks are never tristated. • 1: DRAMs ODT pins are tristated when DRAM clock is stopped or tristated. Note: ODT is not tristated when global tristate flow is disabled (DCBR.TRISTDIS = 1)

Bit Range	Default & Access	Field Name (ID): Description
4:3	1h RW	**Clock Stop/Tristate Enable (ENCKSTP):** Enable/Disable CK Stop/Tristate During Power down. • 00: Disable CK Stop/Tristate During Power down. • 01: Enable CK Stop During Power down. • 10: Enable CK Tristate During Power down. • 11: Reserved Note: CK is not stopped or tristated when global tristate flow is disabled (DCBR.TRISTDIS = 1).
2:1	0h RW	**Low Power Mode Opcode (LPMODEOP):** D-Unit will send the value in this register after it has entered Powerdown Mode and has stopped/tristated the clock. 00: Disable LPMode. Note: LPMODE entry is not possible when global tristate flow is disabled (DCBR.TRISTDIS = 1).
0	0h RW	**Disable Power Down (DISPWRDN):** Setting this bit to 1 disables dynamic control of DRAM Power-Down entry and exit by keeping the CKE pins driven high. BIOS may set it to 1 during DRAM initialization and DDRIO training. This bit should be set to 0 for normal operation. • 0: The D-Unit dynamically controls the CKE pins to place the DRAM devices in Power Down mode and bring them out of Power Down mode. • 1: The D-Unit constantly drives the CKE pins high to keep the DRAM devices from entering Power Down mode when ranks are idle. Note: This bit is overridden if CKEMODE = 1. This bit does not control CKE behavior on SR entry/exit.

3.15 DRAM Refresh Control (D_CR_DRFC) — Offset 1038h

Specifies the parameters to control scheduling of refresh commands.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1038h	20001750 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	1h RW	**Maintenance Operation Channel Block Time (MNTCHNBLKTIME):** Specifies the amount of time that D-unit will block the channel before any DRAM maintenance operation is issued. (in SPIDclk)
28:25	0h RW	**Maintenance Operation Delay CAS Count (MNTDLYCASCOUNT):** When a critical read request is pending in RPQ and a maintenance operation (MRR, ZQCal, Ref, etc.) needs to be performed, D-unit delays the maintenance operation and allows this many read or write requests to be scheduled before allowing the maintenance operation. Note: This mode does not apply to Panic refreshes.
24:22	0h RO	**Reserved (RSVD24_22):** Reserved
21	0h RW	**Disable Refresh Debt Clear (DISREFDBTCLR):** When set, D-Unit will not clear refresh debt before Self Refresh SR Entry: • 0: D-Unit sends all postponed REF commands to DRAM before it enters Self Refresh. • 1: D-Unit enters SR without clearing the Refresh Debt (for Debug only).

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**Refresh Skew Disable (REFSKWDIS):** Disables Skewing of Refresh Counting between Ranks. Each rank has its own refresh counter. By default incrementing these refresh counters are skewed by 1/2 the tREFI period. Setting this bit to a 1 disables this feature and all refresh counters will increment at the same time per tREFI period. Skewing the tREFI counters can improve performance since traffic to all ranks does not have to be blocked to perform refresh. • 0: Incrementing the refresh counters are skewed by 1/2 tREFI period. • 1: All refresh counters will increment at the same time per tREFI period.
19:18	0h RO	**Reserved (RSVD19_18):** Reserved
17:16	0h RO	**Reserved (RSVD17_16):** Reserved
15	0h RW	**Extra Refresh Debit (EXTRAREFDBT):** When set to 1, D-Unit adds one extra refresh debit (for a total of two) on Self-refresh exit.
14:12	1h RW	**Minimum Refresh Rate (MINREFRATE):** Ensures that refresh rate never drops below a certain limit regardless of TQ polling. • 000: Stop issuing refresh commands and accumulating refresh debits. (does not stop tREFI counter). • 001: 0.25x refresh rate (i.e. 4x tREFI same as no limit). • 010: 0.5x refresh rate (i.e. 2x tREFI). • 011: 1x refresh rate (i.e. 1x tREFI). • 100: 2x refresh rate (i.e. 0.5x tREFI). • 101: 4x refresh rate (i.e. 0.25x tREFI). • 110: 4x refresh rate with derating forced on i.e. 0.25x tREFI. • 111: Reserved.
11:8	7h RW	**Refresh Panic Watermark (REFWMPNC):** When the Refresh counter per rank is greater than this value, the D-Unit will send a REF command to the rank regardless of pending requests. Note: REFWMPNC must be greater than or equal to REFWMHI and greater than 2, Max Value must be less than 8 to not violate 9xtREFI JEDEC requirement.
7:4	5h RW	**Refresh High Watermark (REFWMHI):** When the Refresh counter per rank is greater than this value, the D-Unit will send a REF command to the rank if there is no critical priority requests in the pending queues. Note: Value must be greater or equal to 1 and less than or equal to REFWMPNC.
3:1	0h RO	**Reserved (RSVD3_1):** Reserved
0	0h RW	**Opportunistic Refresh Disable (OREFDIS):** Disable opportunistic scheduling of refresh. • 0: D-Unit will send a REF command only if there is no pending request to that rank. • 1: D-Unit will not send any opportunistic refreshes. Refresh commands are only sent when the refresh counter is greater than REFWMHI. Note: When set, DISREFDBTCLR must also be set to be able to enter SR.

3.16 D-Unit Scheduler Control (D_CR_DSCH) — Offset 103Ch

Specifies parameters to control scheduling of commands to DRAM.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 103Ch	3901C08 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Early Write DQ Enable (EARLY_DQ_EN):** Enables D-unit to send the write data to the PHY one clock earlier than WL value (WL - 1). Note: Applicable to DDR3L and DDR4 only.
30:29	0h RW	**BGF Early Read Data Valid (BGF_EARLY_RDDATA_VALID):** Specifies the number of clocks the D-Unit sends the read data valid through the BGF earlier as compared to the data. • 00: Always write read valid in same SPID clock as data (STATIC_0). • 01: Always write read valid one SPID clock before data (STATIC_1). • 10: Write read valid up to 2 SPID clocks before data (DYNAMIC). • 11: Reserved
28:27	0h RW	**SPID Early Read Data Valid (SPID_EARLY_RDDATA_VALID):** Specifies the delay in SPID clocks from RDDATA_VALID assertion to actual data on SPID. The value should match what is programmed in DDRIO (PHY).
26:21	1Ch RW	**Write Pending Queue Count (WPQCOUNT):** Used to limit the number of available slots in Write Pending Queue/ Write Data Buffer. WPQCOUNT will only recognize changes when PMI ISM is not active.
20:16	10h RW	**Read Pending Queue Count (RPQCOUNT):** Used to limit the number of entries in Read Pending Queue. RPQCOUNT will only recognize changes when PMI ISM is not active.
15	0h RO	**Reserved (RSVD15):** Reserved
14:10	7h RW	**Read Return Data Additional Credits (BLKRDBF_ADD_RDDATA_CR):** Number of additional full cacheline (64B) read data return credits exposed to D-Unit when BLKRDBF is set. Note: The value in this field has no effect on Read return credits when BLKRDBF is not set.
9:8	0h RW	**In-Order Mode (INORDERMODE):** • 0h: In order mode disabled: Commands are sent out of order. • 1h: Partial in order mode: Read and Write CAS commands are sent in the order they were received. ACT and PRE can go out of order. • 2h: Full in order mode serialized test: All DRAM commands CAS ACT PRE associated with a PMI request are issued to DDR before any DRAM commands for a subsequent PMI request. • 3h: Reserved. In order modes should be enabled during init/training/CPGC testing. Should never be changed while the D-Unit queues are nonempty. For WIO2, when in-order mode is enabled (01 or 10), D_CR_DSCH_BYPASSEN must be set to 0
7	0h RW	**Idle Bypass Mode Enable (BYPASSEN):** When set new page hit/empty read requests will bypass D-Unit pipeline stages to save latency 0 - Disable Idle Bypass 1 - Enable Idle Bypass Note: Only applies to WIO2, this bit is reserved for other technologies
6	0h RW	**Block When RDB Full (BLKRDBF):** When set D-Unit stops scheduling new read commands to DRAM when the read data buffer (RDB) is full.
5:4	0h RW	**Stretch Mode (STRETCHMODE):** When stretch mode is enabled, commands are initiated only on Phase 0 of SPIDClk. • 00: Stretch mode is disabled. • 01: Commands are initiated on Phase 0 of every SPID clocks. • 10: Commands are initiated on Phase 0 of even SPID clocks. • 11: Commands are initiated on Phase 0 of odd SPID clocks.
3:0	8h RW	**Masked Write Turnaround Delta (TMWR_TA_DELTA):** The value in this register is subtracted from Masked Write to Read, Masked Write to Write and Masked Write to Masked Write turnaround times to account for half BL MWr commands in LPDDR4 and WIO2. • LPDDR4: = MWr tCCD = MWr BL/2 = 8. • WIO2: NA.

3.17 DRAM Calibration Control (D_CR_DCAL) — Offset 1040h

Specifies parameters to control ZQ Calibration.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1040h	1057 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**ZQ Calibration Type (ZQCALTYPE):** Determines whether the ZQ Calibration is a long or short calibration command (due to ZQCALSTRT). 0: Short calibration (ZQCS). 1: Long calibration (ZQCL).
30	0h RW/V	**ZQ Calibration Start Rank 1 (ZQCALSTRTR1):** Set this bit to 1 to start the ZQ calibration sequence on Rank 1. This bit will remain a 1 until the ZQ calibration is complete for rank 1, then it will return to 0. 0: ZQ calibration is done. 1: ZQ calibration has started and is in progress.
29	0h RW/V	**ZQ Calibration Start Rank 0 (ZQCALSTRTR0):** Set this bit to 1 to start the ZQ calibration sequence on Rank 0. This bit will remain a 1 until the ZQ calibration is complete for rank 0, then it will return to 0. 0: ZQ calibration is done. 1: ZQ calibration has started and is in progress.
28:23	0h RO	**Reserved (RSVD28_23):** Reserved
22:21	0h RW	**Self-Refresh Exit ZQ Calibration Control (SRXZQC):** • 00: On DynSR exit ZQ timer determines the ZQ type. When the state is lost (i.e due to AutoPG/S0ix) ZQCL is always performed. • 01: Always perform ZQCL after self refresh exit. In LPDDR4, ZQ with traffic blocked. • 10: Always perform ZQCS on SR exit. For LPDDR4, ZQ while traffic is allowed. • 11: No ZQCL commands are sent (it disables ZQCAL commands on SR exit).
20:18	0h RO	**Reserved (RSVD20_18):** Reserved
17	0h RW	**ZQ Calibration Mode (ZQCLMODE):** Specifies how ZQCal commands are sent to different ranks. • 0: ZQCal commands are sent in parallel to all ranks. • 1: ZQCal commands are sent serially to each rank.
16	0h RW	**Periodic ZQ Calibration Disable (ZQCDIS):** • 0: Periodic ZQ Calibration is Enabled. • 1: Disable periodic ZQ Calibration.
15:14	0h RO	**Reserved (RSVD15_14):** Reserved
13:0	1057h RW	**ZQ Calibration Interval (ZQINT):** Specifies the time interval between two ZQCS (LPDDR3/DDR3L/DDR4) or ZQ Start (LPDDR4) commands to a DRAM device. (in RTC 32.8KHz clocks)

3.18 DRAM Mode Registers Shadow Copy (D_CR_MR_SHADOW) — Offset 1048h

This register contains a copy of Mode Regsiters in the DRAM so that D-unit can only modify certain bits without affecting other values.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1048h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:16	0h RW	**MR Value 2 (MR_VALUE2):** MR3 Shadow Register (DDR4): BIOS writes the correct value of MR3 register in DDR4 into this field at boot time. D-Unit modifies one bit and rewrites the MR3 into DDR4 to enter and exit MPR mode.
15:14	0h RO	**Reserved (RSVD15_14):** Reserved
13:0	0h RW	**MR Value (MR_VALUE):** MR3 Shadow Register (WIO2): BIOS sets the value of this field at boot time based on the DRAM device configuration. D-Unit merges the value in MR3_THERM_OFFSET with this field and writes the result into DRAM MR3 MR2 Shadow Register (DDR3L): BIOS writes the correct value of MR2 register in DDR3L into this field at boot time. D-Unit modifies one bit and rewrites the MR2 into DDR3L DRAM before SR entry. MR4 Shadow Register (DDR4): BIOS writes the correct value of MR4 register in DDR4 into this field at boot time. D-Unit modifies one bit and rewrites the MR4 into DDR4 DRAM after temperature read out.

3.19 VNN Scaling Timer Control (D_CR_VNNTIMER) — Offset 104Ch

Specifies parameters for VNN Scaling Timer in D-Unit. The values in this register will be set by P-code during VNN scaling period.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 104Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**VNN Scaling Timer Enable (VNN_TIMER_EN):** • 0: The D-Unit VNN Scaling Timer is disabled. • 1: The D-Unit VNN Scaling Timer is enabled.
30:12	0h RO	**Reserved (RSVD30_12):** Reserved
11:0	0h RW	**VNN Timer Time (VNN_TIMER_TIME):** The final timer value (in 16 x DRAM clocks).

3.20 Periodic DRAM Temperature Polling Control (TQ) (D_CR_TQCTL) — Offset 1050h

Specifies the control for periodic temperature monitoring and control of DRAM device.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1050h	6C000008 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	3h RW/V	**TQ Data Rank 1 (TQDATAR1):** If Rank 1 is disabled, this value will remain zero. This field contains the data of the last temperature sensor read from Rank 1 DRAM Mode Register. It is overwritten with each command.
28:26	3h RW/V	**TQ Data Rank 0 (TQDATAR0):** This field contains the data of the last temperature sensor read from Rank 0 DRAM Mode Register. It is overwritten with each command.
25:22	0h RO	**Reserved (RSVD25_22):** Reserved
21:8	0h RW	**TQ Poll Period (TQPOLLPER):** This sets the frequency by which the D-Unit initiates temperature sensor read from DRAM mode register to determine required refresh rate (in 4x tREFI units).
7	0h RW	**Temperature Controlled Refresh Range Enable (DDR4 Only) (TCRREN):** When set, after a DRAM temperature read out (MPR sequence), D-unit writes a 1 to bit 2 of MR_SHADOW.MR_VALUE when temperature sensor read out for that rank indicates a value higher than 0x1, and writes a 0 to that bit otherwise. The new MR_VALUE is then written into MR4 of DDR4 for each enabled rank.
6	0h RW/V	**TQ Poll Start Rank 1 (TQPOLL_START_R1):** When set Dunit will initiate temperature sensor read for Rank 1. Hardware will clear the bit once the MR read command is issued for this rank. Note: TQPOLLEN mus be 0 before this bit is set.
5	0h RW/V	**TQ Poll Start Rank 0 (TQPOLL_START_R0):** When set Dunit will initiate temperature sensor read for Rank 0. Hardware will clear the bit once the MR read command is issued for this rank. Note: TQPOLLEN mus be 0 before this bit is set.
4	0h RW	**Self Refresh Temperature Range Enable (DDR3 Only) (SRTEN):** When set, before every Self refresh entry, D-Unit writes a 1 to bit 7 of MR_SHADOW.MR_VALUE when TQDATA for that rank indicates a value higher then 0x3, and writes a 0 to that bit otherwise. The new MR_VALUE is then written into MR2 of DDR3 for each enabled rank.
3	1h RW	**Enable Dynamic Timing Derating (ENDERATE):** When set to 1, the Dynamic Timing Derating is enabled. When the D-Unit determines (via TQ polling) that the DRAM requires timing derating in addition to refresh interval adjustment, the D-Unit will automatically adjust the relevant timing parameters.
2	0h RW	**Enable TQ Data Push (TQDATAPUSHEN):** When set to 1, D-Unit pushes the data form the last temperature sensor read to a punit register.
1	0h RW	**Enable TQ Poll on Self-Refresh Exit (TQPOLLSREN):** This bit enables temperature sensor read on Self Refresh Exit. If disabled, D-Unit will not initiate MR Read sequence to read temperature value on Self-Refresh exit.
0	0h RW	**Enable Periodic TQ Poll (TQPOLLEN):** This bit enables periodic TQ Poll. If disabled, D-Unit will not read the DRAM's temperature sensor value periodically.

3.21 Temperature Offset Control (D_CR_TQOFFSET) — Offset 1054h

Specifies temperature offset and refresh rate adjustments requested by software.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1054h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD31_16):** Reserved
15:11	0h RO	**Reserved (RSVD15_11):** Reserved
10:8	0h RW	**MR4 Adder (MR4_ADDER):** D-Unit adds the value of this field to TQDATA read from MR4 the resulting value is used to control refresh rate and AC timing derating.
7:3	0h RO	**Reserved (RSVD7_3):** Reserved
2	0h RW/V	**MR3 Offset Update (MR3_OFFSET_UPDATE):** When set, D-Unit writes the merged value of MR3_VALUE and MR3_THERM_OFFSET into MR3 of DRAM. D-Unit clears this bit once the value is written.
1:0	0h RW	**MR3 Thermal Offset (MR3_THERM_OFFSET):** (WIO2 only) Everytime MR3_OFFSET_UPDATE is set, D-Unit merges the value in this field with MR3_VALUE and writes the result to MR3 in DRAM

3.22 D-Unit Control Operations (D_CR_DCO) — Offset 1058h

Specifies D-Unit initialization and control operation.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1058h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/V	**Initialization Complete (IC):** Indicates that initialization of the D-Unit has been completed. Memory accesses are permitted and maintenance operation begins. Until this bit is set to a 1, the memory controller will not accept DRAM requests from the Bunit/GSA/2LM (PMI ISMs will not leave idle). Note: Set this bit to 1 only when all other D-Unit registers have been configured. Usually set at the last configuration step by BIOS on cold/warm reset. D-Unit hardware sets this bit on SR exit.
30	0h RO/V	**DDRIO PHY Initialization Complete (DIOIC):** Status indication that the DDRIO PHY initialization is complete reflects the status spid_init_complete signal.
29	0h RO	**Reserved (RSVD29):** Reserved
28	0h RW	**PMI Control Select (PMICTL):** • 0: D-Unit PMI is connected to Bunit/GSA/2LM. • 1: D-Unit PMI is connected to CPGC. Note: D_CR_DSCH_BYPASSEN must be set to 0 in CPGC mode. Note: PMI must be idle and D-Unit BGF_RUN = 0 before changing the value in this register.
27:8	0h RO	**Reserved (RSVD27_8):** Reserved
7:4	0h RO	**Reserved (RSVD7_4):** Reserved
3	0h RW	**Enable PSMI Mode (PSMIEN):** When enabled, D-Unit will synchronize clock crossing signals. • 0: PSMI Mode is disabled. • 1: PSMI Mode is enabled. Note: Change only allowed when D-Unit is idle.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Maintenance Reset (MNTRST):** Writing a 1 to this field resets all maintenance timers. Clears all states and also clears refresh debt queues. This bit must be cleared by software after at least 3 SPID clocks.
1	0h RW	**Enable Maintenance Operations (MNTEN):** Setting this field to 1 enables all maintenance operations. When DCO.IC is set, the maintenance operations are enabled irrespective of the value of this field.
0	0h RO	**Reserved (RSVD0):** Reserved

3.23 Data Scrambler (D_CR_SCRAMCTRL) — Offset 10A4h

Specifies parameters to control data scrambling in D-Unit.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 10A4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Enable Data Scrambler (SCRM_EN):** When set to 1, data scrambling is enabled. When set to 0, data scrambling is disabled. Should be set before D_CR_BGF_CTL_BGF_RUN is set to 1.
30	0h RO	**Reserved (RSVD30):** Reserved
29:28	0h RW	**Scrambler Clock Gate Select (CLOCKGATE):** This field controls how the scrambler output code is clock gated to reduce power. • 00: Clock gate disabled. • 01: Clock Gate every 2 cycles. • 10: Clock Gate every 3 cycles. • 11: Clock Gate every 4 cycles.
27:16	0h RO	**Reserved (RSVD27_16):** Reserved
15:0	0h RW	**Scrambling Key (KEY):** Sets the key for the scrambler. The key should be a random value that is set following each cold boot.

3.24 Error Injection Address Register (D_CR_ERR_INJ) — Offset 10ACh

Contains the target address for ECC error injection.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 10ACh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD31):** Reserved
30:1	0h RW	**Error Injection Target Address (ADDRESS):** Specifies the PMI address of the write transaction to be injected with the error. Only applicable to Write transactions. Read/ under-fill read of the partial write operation is not affected.
0	0h RO	**Reserved (RSVD0):** Reserved

3.25 Error Injection Control Register (D_CR_ERR_INJ_CTL) — Offset 10B0h

Controls injecting correctable or uncorrectable errors into the write requests specified by target address.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 10B0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RSVD31_4):** Reserved
3	0h RW	**Error Injection Type Higher 32B (SEL_HI):** 0 - Uncorrectable Error (UE) is armed for write address matching to inject UE by using the same poisoning scheme, i.e. inverting corresponding write ECC[6:0] on every QW of the 32B data. 1 - Correctable Error (CE) is armed for write address matching to inject CE by inverting corresponding write ECC[0] on every QW of the 32B data.
2	0h RW	**Error Injection Enable Higher 32B (EN_HI):** When set the error injection is continuously armed for higher 32B of D_CR_ERR_INJ_ADDR write address matching until it is cleared.
1	0h RW	**Error Injection Type Lower 32B (SEL_LO):** 0 - Uncorrectable Error (UE) is armed for write address matching to inject UE by using the same poisoning scheme, i.e. inverting corresponding write ECC[6:0] on every QW of the 32B data. 1 - Correctable Error (CE) is armed for write address matching to inject CE by inverting corresponding write ECC[0] on every QW of the 32B data.
0	0h RW	**Error Injection Enable Lower 32B (EN_LO):** When set, the error injection is continuously armed for lower 32B of D_CR_ERR_INJ_ADDR write address matching until it is cleared.

3.26 Error Log Register (D_CR_ERR_ECC_LOG) — Offset 10B4h

Detected ECC errors are captured in this register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 10B4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/V	**CLEAR:** Setting this bit to one clears all fields in this register, including itself.
30:29	0h RW	**PMI VISA Byte Select (ECC_VISA):** Select ECC or PMI byte on VISA : • 00: ECC byte, • 01: PMI Data Byte [7:0], • 10: PMI Data Byte [63:56], • 11: PMI Data Byte [255:248]
28	0h RW/V	**Correctable Single-bit Error (CERR):** This bit is set when a correctable single-bit error occurs on a memory read data transfer. When this bit is set, the address that caused the error and the error syndrome are also logged and they are locked to further single bit errors, until this bit is cleared. A multiple bit error that occurs after this bit is set will override the address/error syndrome information.
27	0h RW/V	**Uncorrectable Multiple-bit Error (MERR):** This bit is set when an uncorrectable multiple-bit error occurs on a memory read data transfer. When this bit is set, the address that caused the error and the error syndrome are also logged and they are locked until this bit is cleared.
26:25	0h RW/V	**Error Burst Number (ERR_BURST):** Burst number (in BL8) of the error within a chunk.
24	0h RW/V	**Error Chunk Number (ERR_CHUNK):** Chunk number of the error. 0 - lower 32B chunk has error if MERR/CERR is set 1 - higher 32B chunk has the error if MERR/CERR is set
23:16	0h RW/V	**Quad Word ECC Syndrome (SYNDROME_QW):** ECC Syndrome for a QW (64 bit) within 32B Address
15:0	0h RW/V	**Request Tag (TAG):** Read Return Tag matches with the PMI Request Tag which triggered the error log.

3.27 D-Unit Fuse Status (D_CR_DFUSESTAT) — Offset 10BCh

Contains the values read from D-Unit fuses.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 10BCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD31_16):** Reserved
15:0	0h RO/V	**D-Unit Fuse Status (FUSESTAT):** D-Unit fuse bits are captured into this register and are available to be read. • [0]: fus_dun_ecc_dis. • [3:1]: fus_dun_max_supported_device_size[2:0]. • [4:4]: fus_dun_lpddr3_dis. • [5:5]: fus_dun_lpddr4_dis. • [6:6]: fus_dun_wio2_dis. • [7:7]: fus_dun_ddr3l_dis. • [8:8]: fus_dun_ddr4_dis. • [15:9]: reserved.

3.28 Major Mode Control (D_CR_MMC) — Offset 1124h

Specifies parameters to control read/write major mode operation and transitions.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1124h	2B01A518 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:27	5h RW	**RAW Conflict Read Priority for WMM Transition (RAW_WMM):** If a conflict read reaches this priority (or greater depending on access class occupancy), WMM will be triggered to unblock the corresponding write. D-Unit will stay in WMM until corresponding write is issued. Note: The value in this bit must not be higher than lowest terminal priority level of each access class.
26	0h RO	**Reserved (RSVD26):** Reserved
25:23	6h RW	**Read Isoch Trigger Priority (RIMPRIO):** If any read in the RPQ is at this programmable priority, RIM is triggered.
22:18	0h RO	**Reserved (RSVD22_18):** Reserved
17:12	1Ah RW	**Write Isoch Threshold (WIMTHRS):** When the number of entries in WPQ is greater than or equal to this value (higher than WMM entry watermark, less than WPQ size), it triggers write isoch mode (WIM).
11:6	14h RW	**Write Major Mode Exit Watermark (WMMEXIT):** When the number of entries in WPQ is less than this value, the D-Unit will switch back to read major mode.
5:0	18h RW	**Write Major Mode Entry Watermark (WMMENTRY):** When the number of entries in WPQ is greater than or equal to this value, the D-Unit will switch to write major mode (WMM). Note: the value must not be set to 0.

3.29 Major Mode RD/WR Counter (Set A and B) (D_CR_MMRDWR_AB) — Offset 1128h

Minimum read and maximum write counter control. This register defines the minimum number of reads in RMM and maximum number of writes in WMM before a mode transition happens.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1128h	1020408 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (RSVD31_26):** Reserved
25:20	10h RW	**Max Writes B (MAXWRB):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set B).

Bit Range	Default & Access	Field Name (ID): Description
19:14	8h RW	**Min Reads B (MINRDB):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set B).
13:12	0h RO	**Reserved (RSVD13_12):** Reserved
11:6	10h RW	**Max Writes A (MAXWRA):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set A).
5:0	8h RW	**Min Reads A (MINRDA):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set A).

3.30 Major Mode RD/WR Counter (Set C and D) (D_CR_MMRDWR_CD) — Offset 112Ch

Minimum read and maximum write counter control. This register defines the minumum number of reads in RMM and maximum number of writes in WMM before a mode transition happens (sets C and D).

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 112Ch	1020408 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (RSVD31_26):** Reserved
25:20	10h RW	**Max Writes D (MAXWRD):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set D).
19:14	8h RW	**Min Reads D (MINRDD):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set D).
13:12	0h RO	**Reserved (RSVD13_12):** Reserved
11:6	10h RW	**Max Writes C (MAXWRC):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set C).
5:0	8h RW	**Min Reads C (MINRDC):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set C).

3.31 Access Class Initial Priority (D_CR_ACCIP) — Offset 1130h

Each field of this register defines the initial priority of one access class.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1130h	17C2 h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD31_15):** Reserved
14:12	1h RW	**Access Class 4 Initial Priority (AC4IP):** Initial priority level of read requests coming with access class 4.
11:9	3h RW	**Access Class 3 Initial Priority (AC3IP):** Initial priority level of read requests coming with access class 3.
8:6	7h RW	**Access Class 2 Initial Priority (AC2IP):** Initial priority level of read requests coming with access class 2.
5:3	0h RW	**Access Class 1 Initial Priority (AC1IP):** Initial priority level of read requests coming with access class 1.
2:0	2h RW	**Access Class 0 Initial Priority (AC0IP):** Initial priority level of read requests coming with access class 0.

3.32 Access Class 0 Priority Promotion Control (D_CR_RD_PROM0) — Offset 1134h

This register defines the priority promotion policy for access class 0. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1134h	1F52940 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
24:20	1Fh RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	Ah RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	Ah RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	Ah RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.33 Access Class 1 Priority Promotion Control (D_CR_RD_PROM1) — Offset 1138h

This register defines the priority promotion policy for access class 1. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the associated level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1138h	14000000 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	Ah RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
24:20	0h RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	0h RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	0h RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	0h RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.34 Access Class 2 Priority Promotion Control (D_CR_RD_PROM2) — Offset 113Ch

This register defines the priority promotion policy for access class 2. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 113Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

Bit Range	Default & Access	Field Name (ID): Description
24:20	0h RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	0h RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	0h RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	0h RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.35 Access Class 3 Priority Promotion Control (D_CR_RD_PROM3) — Offset 1140h

This register defines the aging policy for access class 3. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1140h	1F29400 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
24:20	1Fh RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	5h RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	5h RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	0h RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.36 Access Class 4 Priority Promotion Control (D_CR_RD_PROM4) — Offset 1144h

This register defines the aging policy for access class 3. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1144h	1F5294A h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
24:20	1Fh RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	Ah RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	Ah RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	Ah RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	Ah RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.37 Deadline Threshold (D_CR_DL_THRS) — Offset 1148h

Specifies when the request with initial priority 0 get promoted to a higher priority level.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1148h	6 h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD31_11):** Reserved
10:0	6h RW	**Deadline Threshold (DEADLINE_THRS):** A requests with initial priority of 0 will exit priority 0 when its deadline is equal or less than this value plus current time. This field does not affect the priority of any requests in access classes with initial priority bigger than 0.

3.38 Major Mode Blocking Rules Control (D_CR_MM_BLK) — Offset 114Ch

This register controls blocking rules enforced in RMM and WMM.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 114Ch	1800 h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD31_25):** Reserved
24	0h RW	**WMM Regular Rule 1 (WMM_REG_R1):** Disable WMM unsafe write page hits block safe write page misses same bank.
23:20	0h RO	**Reserved (RSVD23_20):** Reserved
19	0h RW	**WMM Priority Rule 4 (WMM_PRIO_R4):** Disable WMM unsafe priority 1 read miss block write hit to same bank. Note: This rule does not block the bank that is being blocked by WMM_PRIO_R3. Priority rules 1, 3 and 4 should be enabled/disabled together.
18	0h RW	**WMM Priority Rule 3 (WMM_PRIO_R3):** Disable WMM unsafe priority 1 write hit block write miss to same bank. Note: This rule does not block the bank that is being blocked by WMM_PRIO_R1. Priority rules 1,3 and 4 should be enabled/disabled together.
17	0h RW	**WMM Priority Rule 2 (WMM_PRIO_R2):** Disable WMM CAS block rule.
16	0h RW	**WMM Priority Rule 1 (WMM_PRIO_R1):** Disable WMM unsafe top priority 1 write miss block write hit same bank. Priority rules 1, 3 and 4 should be enabled/disabled together.
15:14	0h RO	**Reserved (RSVD15_14):** Reserved
13	0h RW	**RMM Regular Rule 6 (RMM_REG_R6):** Disable RMM unsafe write page hits block safe write page misses same bank.
12	1h RW	**RMM Regular Rule 5 (RMM_REG_R5):** Disable RMM unsafe read page miss block all safe and unsafe write page hit to the same bank. Note: This field must not be set to 0 (enabled) if RMM_REG_R4 is also 0.
11	1h RW	**RMM Regular Rule 4 (RMM_REG_R4):** Disable RMM unsafe write page hit block safe read page miss same bank. Note: This field must not be set to 0 (enabled) if RMM_REG_R5 is also 0.
10	0h RW	**RMM Regular Rule 3 (RMM_REG_R3):** Disable RMM unsafe read page hit block safe read and write page miss same bank. Note: This rule does not block the bank that is being blocked by RMM_PRIO_R3 and RMM_PRIO_R1.
9	0h RW	**RMM Regular Rule 2 (RMM_REG_R2):** Disable RMM unsafe read page empty block safe write page empty same rank.
8	0h RW	**RMM Regular Rule 1 (RMM_REG_R1):** Disable RMM unsafe read page hit block safe write page hit same rank.
7:4	0h RO	**Reserved (RSVD7_4):** Reserved
3	0h RW	**RMM Priority Rule 4 (RMM_PRIO_R4):** Disable RMM unsafe critical read miss block read and write hit to same bank. Note: This rule does not block the bank that is being blocked by RMM_PRIO_R3. Priority rules 1, 3 and 4 should be enabled/disabled together.
2	0h RW	**RMM Priority Rule 3 (RMM_PRIO_R3):** Disable RMM unsafe critical read hit block read and write miss to same bank. Note: This rule does not block the bank that is being blocked by RMM_PRIO_R1. Priority rules 1, 3 and 4 should be enabled/disabled together.
1	0h RW	**RMM Priority Rule 2 (RMM_PRIO_R2):** Disable RMM CAS block rule.
0	0h RW	**RMM Priority Rule 1 (RMM_PRIO_R1):** Disable RMM unsafe top critical read miss block read and write hit same bank. Note: Priority rules 1, 3 and 4 should be enabled/disabled together.

3.39 DRAM Self-Refresh Command (D_CR_DRAM_SR_CMD) — Offset 1154h

Self refresh command register to allow sending WAKE and SUSPEND messages to D-Unit. (Only one bit can be set at a time). Posted writes to this register are not completed until hardware clears the field.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1154h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RSVD31_4):** Reserved
3	0h RW/V	**SUSPENDP:** A SUSPENDP message will put the DRAM into self-refresh mode. The D-Unit will complete servicing outstanding memory requests and flush all queued Refresh commands to DRAM before putting the DRAM in self refresh mode. Finally, a PM message will be sent to the PHY. The bit is cleared by hardware after the PHY indicates the transition requested in the PM message has been completed. D-Unit will perform an MRW to MR17 with an opcode as defined by DPMC0.PASR before it places the DRAM into Self-Refresh.
2	0h RW/V	**SUSPEND:** A SUSPEND message will put the DRAM into self-refresh mode. The D-Unit will complete servicing outstanding memory requests and flush all queued Refresh commands to DRAM before putting the DRAM in Self Refresh mode. Finally, a PM message will be sent to the PHY. The bit is cleared by hardware only after the PHY indicates the transition requested in the PM message has been completed. Note: When COLDWAKE is set prior of setting this bit the DRAM will not be placed in SR.
1	0h RO	**Reserved (RSVD1):** Reserved
0	0h RW/V	**WAKE:** Take PHY out of PM states and wakes the DRAM out of self refresh mode. The bit is cleared by hardware only when the DRAM has exited out of self refresh mode and is accessible. Note: When COLDWAKE is set prior of setting this bit the D-Unit will not send SR exit command and will not set the DCO.IC bit.

3.40 DQS Retraining Control (D_CR_DQS_RETRAINING_CTL) — Offset 1180h

LPDDR4 DQS Retraining control register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1180h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DQS Periodic Retraining Interval (DQS_RETRAIN_INT):** This sets the frequency by which the D-Unit initiates periodic retraining (in 1x NREFI).
15:14	0h RO	**Reserved (RSVD15_14):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:4	0h RW	**DQS Oscillator Runtime (DQS_OSC_RT):** After D-Unit starts DQS oscillator, it must wait this amount of time before being able to read the value in MR18 and MR19 (in 16x DRAM clocks). Value in this register must be at least equal to DRAM's MR23 value. + tOSCO.
3	0h RW/V	**DQS Retrain Start Rank 1 (DQS_RETRAIN_START_R1):** When set, Dunit will initiate LPDDR4 DQS training sequence on Rank1. Hardware will clear bit when sequence completes (after issuing MR19). Note: DQS_RETRAIN_EN must be 0 before setting this bit.
2	0h RW/V	**DQS Retrain Start Rank 0 (DQS_RETRAIN_START_R0):** When set, Dunit will initiate LPDDR4 DQS training sequence on Rank0. Hardware will clear bit when sequence completes (after issuing MR19). Note: DQS_RETRAIN_EN must be 0 before setting this bit.
1	0h RW	**DQS Retrain SRX Exit (DQS_RETRAIN_SRX_EN):** Enable retraining on SR exit. This bit enables LPDDR4 DQS retraining on Self Refresh Exit. If disabled, D-Unit will not perform retraining on SR exit.
0	0h RW	**DQS Retrain Enable (DQS_RETRAIN_EN):** Periodic retraining enable: This bit enables periodic DQS retraining. If disabled, D-Unit will not perform retraining periodically. Note: Will be enabled only if DCO.IC is set and refreshes are enabled in DRF.MINREFRATE.

3.41 MR4 De-Swizzle Control (D_CR_MR4_DESWIZZLE) — Offset 1184h

Controls the data bits swizzling crossbar for MR4.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1184h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD31):** Reserved
30:28	0h RW	**MR4 Bit 2 Select 2nd Byte (MR4_BIT2_SEL2):** Selects bit 2 of MR4 data.
27	0h RO	**Reserved (RSVD27):** Reserved
26:24	0h RW	**MR4 Bit 1 Select 2nd Byte (MR4_BIT1_SEL2):** Selects bit 1 of MR4 data
23	0h RO	**Reserved (RSVD23):** Reserved
22:20	0h RW	**MR4 Bit 0 Select 2nd Byte (MR4_BIT0_SEL2):** Selects bit 0 of MR4 data.
19:18	0h RO	**Reserved (RSVD19_18):** Reserved
17:16	0h RW	**MR4 Byte 2 Select (MR4_BYTE_SEL2):** Selects byte position of the MR4 data for second device.
15	0h RO	**Reserved (RSVD15):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
14:12	0h RW	**MR4 Bit 2 Select (MR4_BIT2_SEL):** Selects bit 2 of MR4 data.
11	0h RO	**Reserved (RSVD11):** Reserved
10:8	0h RW	**MR4 Bit 1 Select (MR4_BIT1_SEL):** Selects bit 1 of MR4 data.
7	0h RO	**Reserved (RSVD7):** Reserved
6:4	0h RW	**MR4 Bit 0 Select (MR4_BIT0_SEL):** Selects bit 0 of MR4 data.
3:2	0h RO	**Reserved (RSVD3_2):** Reserved
1:0	0h RW	**MR4 Byte Select (MR4_BYTE_SEL):** Selects byte position of the MR4 data first device.

3.42 DRAM Rank Population 0 (D_CR_DRP0) — Offset 1200h

Rank configuration register.

This register is start of dunit 9 LPDDR4 Channel 2.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1200h	10000000 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RW	**DRAM Device Per Rank (DRAMDEVICE_PR):** Specifies the number of DRAM devices that are ganged together to form a single rank. • 00: 1 DRAM device in each rank. • 01: 2 DRAM devices in each rank. • 10: 4 DRAM devices in each rank. • 11: 8 DRAM devices in each rank. Note: The actual number of devices is one more than the value programmed when ECC is enabled.
29:28	1h RW	**Address Decode (ADDRDEC):** Specifies the address mapping to be used: • 00: 1KB (A). • 01: 2KB (B). • 10: 4KB (C). • 11: Reserved.
27:25	0h RW	**Burst Length Mode (BLMODE):** • 000: Fixed BL8. • 001: Onthefly BL8. • 010: Fixed BL16. • 011: Onthefly BL16. • 100: Fixed BL32. • 101: Onthefly BL32. • 110-111: Reserved.

Bit Range	Default & Access	Field Name (ID): Description
24:22	0h RW	**DRAM Type (DRAMTYPE):** • 000: Reserved. • 001: Reserved. • 010: LPDDR4. • 011: Reserved. • 100: DDR4. • 101-111: Reserved. Note: The D-Unit should only use this field if allowed by fuse.
21	0h RW	**ECC Enable (ECCEN):** • 0: ECC is disabled. • 1: ECC is enabled. This bit determines if the D-Unit treats the PMI BE_ECC bits as ECC bits or Byte Enables. The D-Unit should not allow this bit to be set if ECC is disabled by fuse. This should only be used in configurations that support ECC.
20:19	0h RW	**CA Swizzle Type (CASWIZZLE):** • 00: uniDIMM/SODIMM/UDIMM. • 01: BGA. • 10: NA. • 11: uniDIMM/SODIMM/UDIMM with Rank 1 mirrored (DDR4 Only).
18:17	0h RO	**Reserved (RSVD18_17):** Reserved
16	0h RW	**Fine Bank Group Interleaving Enable (FBGINTEN):** When enabled, D-unit will use a lower order address bit in the bank hashing function (DDR4 Only). • 0: Bank Group Interleave disabled. • 1: Bank Group Interleave enabled. Note: BAHEN must be set for this bit to take effect.
15	0h RW	**Bank Address Hashing Enable (BAHEN):** • 0: Bank Address Hashing disabled. • 1: Bank Address Hashing enabled.
14	0h RW	**Rank Select Interleave Enable (RSIEN):** • 0: Rank Select Interleaving disabled. • 1: Rank Select Interleaving enabled.
13:9	0h RO	**Reserved (RSVD13_9):** Reserved
8:6	0h RW	**DRAM Device Density (DDEN):** Density of the DRAM devices populated on Ranks 0 and 1. • 000: 4 Gb. • 001: 6 Gb. • 010: 8 Gb. • 011: 12 Gb. • 100: 16 Gb. • 101-111: Reserved. Note: For LPDDR4 this value is the die density.
5:4	0h RW	**DRAM Device Data Width (DWID):** Data width of the DRAM device populated on Ranks 0 and 1. • 00: x8. • 01: x16. • 10: x32. • 11: x64.
3	0h RO	**Reserved (RSVD3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Dual Data Mode Enable (DDMEN):** • 0: PMI Dual Data Mode is disabled in D-Unit. • 1: PMI Dual Data Mode is enabled. Note: Dual Data Mode must be enable for DDR3L/DDR4 configurations.
1	0h RW	**Rank Enable 1 (RKEN1):** Enable Rank 1: Must be set to 1 to enable use of this rank.
0	0h RW	**Rank Enable 0 (RKEN0):** Enable Rank 0: Must be set to 1 to enable use of this rank. Note: Setting this bit to 0 is not a functional mode.

3.43 DRAM Timing Register 9A (D_CR_DTR9A) — Offset 1204h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1204h	14A546 h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RO	**Reserved (RSVD31_23):** Reserved
22:18	5h RW	**Write to Write DQ Delay Same Bank Group Same Rank (TWWSR_L):** Specifies the delay from a DRAM Write to another Write command within the same bank group of the same rank (in DRAM clocks). DDR4 Equation: tWWSR_L = tCCD_L.
17:13	5h RW	**Read to Read DQ Delay Same Bank Group Same Rank (TRRSR_L):** Specifies the delay from a DRAM Read to another Read command within the same bank group of the same rank (in DRAM clocks). DDR4 Equation: tRRSR_L = tCCD_L.
12:6	15h RW	**Write to Read DQ Delay Same Bank Group Same Rank (TWRSR_L):** Specifies the delay from a DRAM Read to Write command within the same bank group of the same rank (in DRAM clocks). • DDR4 Equation: tWRSR_L = CWL + tDQSSmax + BL/2 + tWPST + tWTR_L.
5:0	6h RW	**Row Activation to Row Activation to Same Bank Group Delay [tRRD_L] (TRRD_L):** Specifies the minimum delay (in DRAM clocks) between two DRAM Activate commands to different banks within the same bank group of a rank. (DDR4 Only)

3.44 DRAM Timing Register 0A (D_CR_DTR0A) — Offset 1208h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1208h	820702CB h

Bit Range	Default & Access	Field Name (ID): Description
31:27	10h RW	**Valid Clocks Before CKE High [tCKCKEH/tCSCKEH/tCKSRX] (TCKCKEH):** Number of valid clocks before CKE high (in DRAM clocks). • LPDDR4: The value in this register covers both tCKCKEH and tCSCKEH. • DDR3L/DDR4/LPDDR3/WIO2: The value covers tCKSRX which is defined as the number of valid DRAM clocks that have to toggle before the issuing of the Self Refresh Exit SRX. This value is also used if the clock frequency is changed or the clock is stopped or tristated during Power Down i.e. the number valid DRAM clocks that have to toggle before the issuing of the Power Down Exit PDX command. TCKCKEH can be used to compensate for clock stabilization delays in the motherboard. Note: D-unit hardware enforces minimum of two SPID clock before CKEH, any value in this register is the additional time.
26:22	8h RW	**Exit Self-Refresh to Valid Commands Requiring a Locked DLL Delay [tXSDLL] (TXSDLL):** D-Unit waits max(tXSR+tZQCL/tZQCS, tXSDLL) before allowing traffic to DRAM (in 32 x DRAM Clocks). LPDDR3/LPDDR4/WIO2: tXSDLL = 0. DDR3L/DDR4: tXSDLL = tDLLK. Note: In the equation above, tZQCL/tZQCS = 0 if no ZQ is performed on SR exit.
21:12	70h RW	**Exit Self-Refresh to Valid Command Delay [tXS/tXSR] (TXSR):** DDR3L/DDR4: tXS - Delay between Self Refresh Exit SRX to any DRAM Command not requiring DLL Lock. LPDDR/WIO2: tXSR - Delay between Self Refresh Exit SRX to any DRAM Command. (in DRAM clocks).
11:6	Bh RW	**Activate RAS to CAS Command Delay [tRCD] (TRCD):** Specifies the delay between a DRAM Activate command and a DRAM Read or Write command to the same bank (in DRAM clocks). Note: Derating adds 1.875ns to this timing.
5:0	Bh RW	**Precharge to Activate Command Delay of a Single Bank [tRPpb] (TRPPB):** Specifies the delay between a DRAM Precharge command and a DRAM Activate command to the same bank (in DRAM Clocks). Note : this CR should be constrained to a minimum of 4 in LPDDR3/DDR3L/WIO2 and minimum of 8 in LPDDR4/DDR4. Note: Derating adds 1.875ns to this timing.

3.45 DRAM Timing Register 1A (D_CR_DTR1A) — Offset 120Ch

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 120Ch	30481218 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	6h RW	**Exit Power Down to Next Command Delay [tXP] (TXP):** Specifies the delay from the DRAM Power Down Exit (PDX) command to any valid command (in DRAM clocks). Note: The value in this field must be programmed to tXPDLL when Slow Exit Mode Power-down is enabled for DDR3L.
26	0h RO	**Reserved (RSVD26):** Reserved
25:14	120h RW	**ZQ (long) Calibration Time [tZQCL/tZQCAL] (TZQCL):** • LPDDR3/DDR3L/DDR4: tZQCL/tZQoper - Specifies the delay between the DRAM ZQ Calibration Long (ZQCL) command and any DRAM command during normal operation. • LPDDR4: tZQCAL - ZQ Calibration time (in DRAM clocks). Note: This field defines the ZQ Calibration Long delay during normal operation. It is not the same as tZQinit which uses the same ZQCL command but the delay is longer. tZQinit applies only during poweron initialization of the DRAM devices and tZQoper applies during normal operation. BIOS executes the DRAM initialization sequence so it has to ensure tZQinit is met and not the D-Unit.

Bit Range	Default & Access	Field Name (ID): Description
13:6	48h RW	**ZQ Short Calibration Time [tZQCS] (TZQCS):** ZQCS to any DRAM Command Delay: Specifies the delay between the DRAM ZQ Calibration Short (ZQCS) command and any DRAM command (in DRAM clocks). DDR3L/DDR4/LPDDR3 only. LPDDR4 does not support ZQCS command
5:0	18h RW	**ZQ Latch Time [tZQLAT] (TZQLAT):** Specifies the delay between the DRAM ZQ Calibration Latch command and any DRAM command (in DRAM clocks). LPDDR4 only. Not used in DDR3L/DDR4/LPDDR3/WIO2.

3.46 DRAM Timing Register 2A (D_CR_DTR2A) — Offset 1210h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1210h	46080C30 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	118h RW	**All Bank Refresh Cycle Time [tRFCab] (NRFCAB):** Specifies the delay between the REFab command to the next valid command. (in DRAM clocks)
21	0h RO	**Reserved (RSVD21):** Reserved
20:17	4h RW	**CKE Minimum Pulse Width [tCKE] (TCKE):** Specifies the minimum time from CKEL to CKEH (in DRAM clocks).
16	0h RO	**Reserved (RSVD16):** Reserved
15:0	C30h RW	**Refresh Interval Time [tREFI] (NREFI):** Specifies the average time between refresh commands. JEDEC Base Refresh Interval time (in DRAM clocks). Note: D-Unit will ignore the 2 LSBs of this field.

3.47 DRAM Timing Register 3A (D_CR_DTR3A) — Offset 1214h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1214h	3002EA28 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	6h RW	**Read to Precharge Delay [tRDPRE] (TRTP):** Specifies the minimum delay between the DRAM Read and Precharge commands to the same bank (in DRAM clocks). • LPDDR3 Equation: = BL/2 + tRTP - 4. • LPDDR4 Equation: = BL/2 + Max (8, tRTP) - 8. • WIO2 Equation: = NA. • DDR3L/DDR4 Equation: = tRTP.

Bit Range	Default & Access	Field Name (ID): Description
26:20	0h RW	**CAS to CAS Command Delay Adder (TCCD_INC):** Specifies the number of clocks to be added to turnaround times (for Stretch Mode). It increases delay between Read to Read or Read to Write commands by the amount programmed in this field (in 4 x DRAM clocks).
19:13	17h RW	**Write to Precharge Command Delay [tWRPRE] (TWTP):** Specifies the minimum delay between the DRAM Write command and the Precharge command to the same bank (in DRAM clocks). • LPDDR3/LPDDR4/WIO2 Equation: tWTP = BL/2 + WL + tWR + 1. • DDR3L/DDR4 Equation: tWTP = BL/2 + CWL + tWR.
12:11	1h RW	**DRAM Command Valid Duration (TCMD):** Specifies the number of DRAM clocks a command is held valid on the DRAM Address and Control buses. 1N is the DDR3 basic requirement. 2N is the extended mode for board signal integrity. • 0h: Reserved. • 1h: 1 DRAM Clock (1N). • 2h: 2 DRAM Clocks (2N). • 3h: Reserved. Note: DDR3L/DDR4 only. tCMD must be set to 1N for LPDDR3/LPDDR4/WIO2.
10:6	8h RW	**Write Latency [WL/CWL] (TCWL):** The delay between the internal write command and the availability of the first word of DRAM input data (in DRAM clocks).
5:0	28h RW	**Write CAS to Masked Write CAS Delay Same Bank (TWMWSB):** Specifies the minimum delay between DRAM Write command to Masked Write command to same bank (in DRAM clocks). • LPDDR4 Equation: tWMWSB = tCCDMW (BL16) or tCCDMW + 8 (BL32). • WIO2 Equation: NA. Note: Masked Write operation in LPDDR4 is always BL16 and in WIO2 is always BL4. D-Unit applies this timing for same rank as well as same bank.

3.48 DRAM Timing Register 4A (D_CR_DTR4A) — Offset 1218h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1218h	30209149 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	30h RW	**Four Bank Activate Window [tFAW] (TFAW):** A rolling timeframe in which a maximum of four Activate commands can be issued to the same rank. This is to limit the peak current draw from the DRAM devices (in DRAM clocks).
23:18	8h RW	**Write to Read DQ Delay Different Ranks (TWRDR):** Specifies the delay from the start of a Write data burst of one rank to the start of a Read data burst of a different rank (in DRAM clocks). • LPDDR3 Equation: tWRDR = WL + tDQSSmax + BL/2 - (RL + tDQSCKmin - tRPRE). • LPDDR4 Equation: tWRDR = WL - RL + BL/2 + 4 - tDQSCKmin. • DDR3L/DDR4 Equation: tWRDR = CWL + tDQSSmax + BL/2 - (CL + tDQSCKmin - tRPRE). Note: For LPDDR3/4 using ODT, this latency may need to be increased by tODToffadj.

Bit Range	Default & Access	Field Name (ID): Description
17:12	9h RW	**Read to Write DQ Delay Different Ranks (TRWDR):** Specifies the delay from the start of a Read data burst of one rank to the Start of a Write data burst of a different rank (in DRAM clocks). • LPDDR3/LPDDR4 Equation: tRWDR = RL + tDQSCKmax + BL/2 - (WL - tWPRE). • DDR3L/DDR4 Equation: tRWDR = CL + tDQSCKmax + BL/2 - (CWL - tWPRE). Note: For LPDDR3/4 using ODT, this latency may need to be adjusted by tODTon. Note: For DDR3L/DDR4 using ODT, this latency may need to be increased by one clock.
11:6	5h RW	**Write to Write DQ Delay Different Ranks (TWWDR):** Specifies the delay from the start of a Write data burst of one rank to the start of a Write data burst of a different rank (in DRAM clocks). • LPDDR3/DDR3L/DDR4 Equation: tWWDR = BL/2 + tDQSSmax - tDQSSmin + tWPRE. • LPDDR4 Equation: tWWDR = BL/2 + 4 - tDQSSmin. Note: For LPDDR3/4 using ODT, this latency may need to be increased by tODToffadj.
5:0	9h RW	**Read to Read DQ Delay Different Ranks (TRRDR):** Specifies the delay from the start of a Read data burst of one rank to the Start of a Read data burst of a different rank (in DRAM clocks). • Equation: tRRDR = BL/2 + tDQSCKmax - tDQSCKmin + tRPRE.

3.49 DRAM Timing Register 5A (D_CR_DTR5A) — Offset 121Ch

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 121Ch	304200C2 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	6h RW	**Row Activation to Row Activation Delay [tRRD] (TRRD):** Specifies the minimum delay in DRAM clocks between two DRAM Activate commands to the same rank but different banks (tRC is the minimum delay between activations of the same bank). Note: Derating adds 1.875ns to this timing.
26	0h RO	**Reserved (RSVD26):** Reserved
25:23	0h RW	**Derate Increment (TDERATE_INC):** Specifies the additional delay that is added to DRAM timing when indicated by MR4 status. (in DRAM clocks) LPDDR3/LPDDR4/WIO2: Value is 1.875ns. Note: The value in this register is only added to these timing parameters: tRCD, tRAS, tRP and tRRD.
22:18	10h RW	**Write to Write DQ Delay Same Rank (TWWSR):** Specifies the delay from a DRAM Write to another Write command of the same rank (in DRAM clocks). Equation: tWWSR = tCCD.
17:13	10h RW	**Read to Read DQ Delay Same Rank (TRRSR):** Specifies the delay from a DRAM Read to another Read command of the same rank (in DRAM clocks). Equation: tRRSR = tCCD.
12:6	3h RW	**Write to Read DQ Delay Same Rank (TWRSR):** Specifies the delay from a DRAM Read to Write command of the same rank (in DRAM clocks). • LPDDR3/LPDDR4/WIO2 Equation: tWRSR = WL + tDQSSmax + BL/2 + tWTR. • DDR3L Equation: tWRSR = CWL + tDQSSmax + BL/2 + tWTR. • DDR4 Equation: tWRSR = CWL + tDQSSmax + BL/2 + tWTR_S.

Bit Range	Default & Access	Field Name (ID): Description
5:0	2h RW	**Read to Write DQ Delay Same Rank (TRWSR):** Specifies the delay from a DRAM Read to a Write command of the same rank (in DRAM clocks). • LPDDR3/LPDDR4/WIO2 Equation: tRWSR = RL + tDQSCKmax + BL/2 - WL + tWPRE. • DDR3L/DDR4 Equation: tRWSR = CL + tDQSCKmax + BL/2 - CWL + tWPRE. Note: For LPDDR3/4 using ODT, this latency may need to be increased by tODToffadj. Note: For DDR3L/DDR4 using ODT, this latency may need to be increased by one clock.

3.50 DRAM Timing Register 6A (D_CR_DTR6A) — Offset 1220h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1220h	20100000 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	20h RW	**Opportunistic Refresh Idle Timer (OREFDLY):** Rank idle period that defines an opportunity for refresh (in DRAM clocks).
23:19	2h RW	**Valid Clocks After CKE Low [tCKELCK/tCKELCS/tCPDED/tCKSRE] (TCKCKEL):** Specifies the amount of time that DRAM clocks need to toggle after CKE goes low (in DRAM Clocks). • For WIO2/LPDDR3, this covers tCPDED. • For LPDDR4, this covers both tCKELCK and tCKELCS. • For DDR3L/DDR4, this is tCKSRE. Note: D-Unit hardware enforces minimum of one SPID clocks after CKEL, any value in this register is the additional time.
18:15	0h RW	**Reserved (RSVD18_15):** Reserved
14:8	0h RW	**Mode Register Read to Any Command Delay (TPSTMRRBLK):** Specifies the quiet time after issuing MRR command (in DRAM Clocks). This is the channel block time for the MR19 command issued as part of LPDDR4 DQS Retraining flow. For all other MR commands this is the rank block time. Note: D-Unit treats MRR as a read and always applies relevant turnaround times, any value programmed in this CR must be greater than those turnaround times for D-Unit to enforce any additional time from MRR to the next read/write.
7	0h RO	**Reserved (RSVD7):** Reserved
6:0	0h RW	**Any Command to Mode Register Read/Write Delay (TPREMRBLK):** Specifies the channel quiet time before issuing MRR/MRW command. (in DRAM clocks). • LPDDR3 Equation: = tXP + tMRRI - 4. • LPDDR4 Equation: = tXP + tMRRI - 8. • DDR3L Equation: = CWL + BL/2 + 1 - 8 • DDR4 Equation: = CWL + BL/2 + 1 - 16. Note: D-Unit treats MRR as a read and MRW/MRS as a write and always applies relevant turnaround times, any value programmed in this CR must be greater than those turnaround times for D-Unit to enforce any additional time from previous read/writes.

3.51 DRAM Timing Register 7A (D_CR_DTR7A) — Offset 1224h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1224h	D060C06 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	3h RW	**All Bank Precharge to Activate Command Delay [tRPab] (TRPAB):** Specifies the delay between a DRAM Precharge All Bank command and a DRAM Activate command (in DRAM Clocks). Note: This CR should be constrained to a minimum of 4 in LP3 and minimum of 8 in LP4. Note: Derating adds 1.875ns to this timing. • For LPDDR, tRPpb = tRP, tRPab = tRP + 3ns. • For DDR3L, DDR4 and WIO2 8ch tRPpb = tRPab = tRP.
25:23	2h RW	**Mode Register Write to any Command Delay [tMRD/tMRW] (TPSTMRWBLK):** Specifies the quiet time after issuing MRW command (in 8 x DRAM clocks). Note: This time covers for both tMRD and tMRW.
22:16	6h RW	**Write Command to Power Down Delay [tWRPDEN] (TWRPDEN):** Specifies the minimum time between a write command to PowerDown command (in DRAM clocks). Must be at least equal to tWR + tCCD + tWL + 2.
15:9	6h RW	**Read Command to Power Down Delay [tRDPDEN] (TRDPDEN):** Specifies the minimum time between a read command to PowerDown command (in DRAM clocks). Must be at least equal to CL/RL + tDQSCKmax + tCCD + 1.
8:7	0h RO	**Reserved (RSVD8_7):** Reserved
6:0	6h RW	**Row Activation Period [tRAS] (TRAS):** Specifies the minimum delay between the DRAM Activate and Precharge commands to the same bank (in DRAM clocks). Note: Derating adds 1.875ns to this timing.

3.52 DRAM Timing Register 8A (D_CR_DTR8A) — Offset 1228h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1228h	CC50A18 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	3h RW	**Minimum Self-Refresh Time [tSR/tCKESR] (TCKESR):** Specifies the minimum time that DRAM should remain in SR (in DRAM clocks).
25:21	6h RW	**Minimum Low Power Mode Residency (LPMDRES):** Specifies the minimum time that PHY should remain in LPMode (in DRAM clocks).
20:15	Ah RW	**Low Power Mode Exit to Clock Enable Delay (LPMDTOCKEDLY):** Specifies the minimum time between the LP Mode exit to the CK stop/tristate deassertion and powerdown exit (in DRAM clocks). Note: Must be equal to t_idle_latency published in the DDRIO PHY and less than 0x3C.
14:8	Ah RW	**Clock Stop to Low Power Mode Delay (CKETOLPMDDLY):** Specifies the time between CK stop/tristate to the Low Power Mode entry. This timing parameter is used to delay Low Power Mode entry (in DRAM clocks). Note: Must be at least equal to t_idle_length parameter published in the DDRIO PHY and less than 0x7C.
7:0	18h RW	**Power Down Idle Timer (PWDDLY):** This is a non-JEDEC timing parameter used to delay powerdown entry (in DRAM clocks).

3.53 D-Unit ODT Control Register A (D_CR_DOCRA) — Offset 122Ch

Specifies the parameters to control DRAM ODT.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 122Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29	0h RW	**Rank 1 Read ODT Control (R1RDODTCTL):** Specifies the behavior of ODT signals when a Read command is issued to Rank 1. 0 - Read ODT is disabled for Rank 1 1 - Assert ODT to for Rank 0 (non-targeted Rank) Note: This register should be set to 0 for LPDDR3 devices
28	0h RW	**Rank 0 Read ODT Control (R0RDODTCTL):** Specifies the behavior of ODT signals when a Read command is issued to Rank 0. 0 - Read ODT is disabled for Rank 0 1 - Assert ODT to for Rank 1 (non-targeted Rank) Note: This register is reserved for LPDDR3 devices
27:26	0h RW	**Rank 1 Write ODT Control (R1WRODTCTL):** Specifies the behavior of ODT signals when a Write command is issued to Rank 1. 00 - Write ODT is disabled 01 - Assert ODT to Rank 0 (non-targeted Rank) 10 - Assert ODT to Rank 1 (targeted Rank) 11 - Assert ODT to Rank 0 and Rank 1 Note: 10 and 11 are reserved values for LPDDR3
25:24	0h RW	**Rank 0 Write ODT Control (R0WRODTCTL):** Specifies the behavior of ODT signals when a Write command is issued to Rank 0. 00 - Write ODT is disabled 01 - Assert ODT to Rank 0 (targeted Rank) 10 - Assert ODT to Rank 1 (non-targeted Rank) 11 - Assert ODT to Rank 0 and Rank 1 Note: 10 and 11 are reserved values for LPDDR3
23:18	0h RO	**Reserved (RSVD23_18):** Reserved
17:14	0h RW	**Read ODT assertion to de-assertion delay (DDR3L/DDR4) (RDODTSTOP):** Specifies Read ODT assertion to ODT de-assert delay (in DRAM clocks). DDR3L Equation: RDODTSTOP = 6. DDR4 Equation: RDODTSTOP = 5 + tRPRE. Note: Add 1 if DOCRx.RDODTSTART = CL - CWL in 2N mode.
13	0h RO	**Reserved (RSVD13):** Reserved
12:9	0h RW	**Read command to ODT assertion delay (DDR3L/DDR4) (RDODTSTART):** Specifies Read ODT assertion delay after Read Command (in DRAM clocks). DDR3L/DDR4 Equation: RDODTSTART = CL - CWL + tWPRE - tRPRE. Note: In DDR3L/DDR4 2N mode add 1 to enable termination at the start of read data burst.
8:5	0h RW	**Write ODT Assertion to De-assertion Delay (WRODTSTOP):** Specifies number of clocks after ODT assertion that D-Unit deasserts ODT signal (in DRAM clocks). LPDDR3 Equation: WRODTSTOP = RU(tODTon(max)/tCK) + RU((tDQSSmax+tWPST)/tCK) + BL/2 - RD(tODToffmin/tCK) DDR3L Equation: WRODTSTOP = 6. DDR4 Equation: WRODTSTOP = 5 + tWPRE. Note: Add 1 if DOCRx.WRODTSTART = 0 in 2N mode.
4	0h RO	**Reserved (RSVD4):** Reserved
3:0	0h RW	**Write command to ODT assertion delay (WRODTSTART):** Specifies number of clocks after Write command that D-Unit asserts ODT signal (in DRAM clocks). LPDDR3 Equation: WRODTSTART = WL - RU(tODTon(max)/tCK) DDR3L/DDR4 Equation: WRODTSTART = 0 Note: In DDR3L/DDR4 2N mode the value can be set to 0 to assert ODT one DRAM clock earlier than the Write Command (WR) or set to 1 to assert at the same clock as command (CS assertion).

3.54 D-Unit Power Management Control 0 (D_CR_DPMC0) — Offset 1230h

Specifies the parameters to control D-Unit power management features.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1230h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**Reserved (RSVD31_29):** Reserved
28:24	0h RW	**SUSPEND/SUSPENDP Power Management Message Opcode (SUSPMOP):** DDRIO PHY Power Mode Opcode: After the D-Unit has placed the DRAM devices in Self Refresh/PASR mode as the result of a SUSPEND/SUSPENDP message, it sends this 5-bit value to the DDRIO PHY to tell it which power saving mode it should enter. Changing this register value while in SUSPEND will have no effect. Note: This opcode cannot be a PM state where it disables PHY PLLs i.e PM7 in LPDDR PHY.
23	0h RO	**Reserved (RSVD23):** Reserved
22	0h RW	**PM Message Wait for Clock Gate Enable (SRPMCLKW):** Specifies when it is safe to send PM message to the PHY. When enabled, D-Unit waits for SPID Clock to deassert before sending a PM message on SR entry. • 0: D-Unit will not wait for SPID_clk to deassert before sending the PM message to PHY. • 1: D-Unit will wait for SPID_clk to deassert before sending PM message to the PHY. Note: The value must be 1 when DYNPMOP = 7h.
21:17	0h RW	**Dynamic Self-Refresh Power Management Message Opcode (DYNPMOP):** DDRIO PHY Power Mode Opcode: After the D-Unit has placed the DRAM devices in Self Refresh mode as the result of a Dynamic Self-Refresh, it sends this 5bit value to the DDRIO PHY to tell it which power saving mode it should enter. Changing this register value while in self-refresh will only change the PM state for the next entry in DynSR.
16	0h RW	**Dynamic Self-Refresh Enable (DYNSREN):** When set to 1, the D-Unit will automatically control DRAM Self Refresh entry and exit based on interface state and requests in pending queues. When there is no pending request in the queues and PMI is idle, then the D-Unit will place the DRAM devices in Self Refresh mode. The DRAM devices will be brought out of Self-Refresh when idle conditions don't hold.
15:0	0h RW	**Self-Refresh Entry Delay (SREDLY):** Specifies the minimum time the D-Unit will wait before it enters Dynamic Self-Refresh mode when idle (in 16x DRAM Clocks). Note: The value in this field needs to be minimum of 4 in functional mode and minimum of 50 in PSMI mode.

3.55 D-Unit Power Management Control 1 (D_CR_DPMC1) — Offset 1234h

Specifies the parameters to control D-Unit power management features.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1234h	10000028 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29	0h RW	**D-Unit Repeaters Clock Gate Disable (RPTCLKGTDIS):** Setting this bit to 0 allows majority of the repeaters between D-Unit and PHY to clock gate when there is no activity in order to save power. • 0 - Enable Repeaters clock gating. • 1 - Disable Repeaters clock gating. Note: This is a de-feature bit and should be set to 0 for normal operation. Note: The value should only change after DRAM Timing Registers (DTR) are programmed.
28	1h RW	**IOSF-SB End Point Clock Gate Disable (SBEPCLKGTDIS):** Setting this bit to 0 enables the clock gating of IOSF-SB End Points in D-Unit and CPGC when there is no IOSF-SB activity in order to save power. • 0 - Enable IOSF-SB EP clock gating. • 1 - Disable IOSF-SB clock gating. Note: This is a de-feature bit and should be set to 0 for normal operation. Note: The value should only change after DRAM Timing Registers (DTR) are programmed.
27	0h RW	**Local Clock Gate Disable (CLKGTDIS):** Setting this bit to 0 allows the majority of the D-Unit clocks to be gated off when there is no activity in order to save power. When set to 1, D-Unit clockgating is disabled. • 0: Enable. • 1: Disable. Note: This is a de-feature bit and should be set to 0 for normal operation. Note: The value should only change after DRAM Timing Registers (DTR) are programmed.
26	0h RW	**Chip Select Tristate Enable (CSTRIST):** • 0: The DRAM CS pins associated with the enabled ranks are never tristated. • 1: The DRAM CS pins are tristated when DRAM clock is stopped or tristated. Note: CS is not tristated when global tristate flow is disabled (DCBR.TRISTDIS = 1).
25:24	0h RW	**Command/Address Tristate (CMDTRIST):** • 00: The DRAM CA pins are never tristated. • 01: The DRAM CA pins are only tristated when all enabled CKE pins are low. • 10: The DRAM CA pins are tristated when not driving a valid command. • 11: Reserved
23:16	0h RW	**Partial Array Self-Refresh Segment Mask (PASR):** This is the Segment Mask used for the MRW to enable PASR during SUSPENDP (Partial Array Self Refresh entry).
15:8	0h RW	**Page Close Timeout Period (PCLSTO):** Specifies the time from the last access of a DRAM page until that page is scheduled to close by sending a Precharge command to DRAM (in 16 x DRAM clocks).
7	0h RW	**Page Close Timeout Disable (PCLSTODIS):** When disabled, D-Unit will not close the DRAM page when idle. • 0: Enable page close timer. • 1: Disable page close timer (Used during DRAM init and DDRIO training).
6	0h RO	**Reserved (RSVD6):** Reserved
5	1h RW	**ODT Tristate Enable (ODTTRIST):** • 0: The DRAM ODT pins associated with the enabled ranks are never tristated. • 1: DRAMs ODT pins are tristated when DRAM clock is stopped or tristated. Note: ODT is not tristated when global tristate flow is disabled (DCBR.TRISTDIS = 1)

Bit Range	Default & Access	Field Name (ID): Description
4:3	1h RW	**Clock Stop/Tristate Enable (ENCKSTP):** Enable/Disable CK Stop/Tristate During Power down. • 00: Disable CK Stop/Tristate During Power down. • 01: Enable CK Stop During Power down. • 10: Enable CK Tristate During Power down. • 11: Reserved Note: CK is not stopped or tristated when global tristate flow is disabled (DCBR.TRISTDIS = 1).
2:1	0h RW	**Low Power Mode Opcode (LPMODEOP):** D-Unit will send the value in this register after it has entered Powerdown Mode and has stopped/tristated the clock. 00: Disable LPMode. Note: LPMODE entry is not possible when global tristate flow is disabled (DCBR.TRISTDIS = 1).
0	0h RW	**Disable Power Down (DISPWRDN):** Setting this bit to 1 disables dynamic control of DRAM Power-Down entry and exit by keeping the CKE pins driven high. BIOS may set it to 1 during DRAM initialization and DDRIO training. This bit should be set to 0 for normal operation. • 0: The D-Unit dynamically controls the CKE pins to place the DRAM devices in Power Down mode and bring them out of Power Down mode. • 1: The D-Unit constantly drives the CKE pins high to keep the DRAM devices from entering Power Down mode when ranks are idle. Note: This bit is overridden if CKEMODE = 1. This bit does not control CKE behavior on SR entry/exit.

3.56 DRAM Refresh Control (D_CR_DRFC) — Offset 1238h

Specifies the parameters to control scheduling of refresh commands.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1238h	20001750 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	1h RW	**Maintenance Operation Channel Block Time (MNTCHNBLKTIME):** Specifies the amount of time that D-unit will block the channel before any DRAM maintenance operation is issued. (in SPIDclk)
28:25	0h RW	**Maintenance Operation Delay CAS Count (MNTDLYCASCOUNT):** When a critical read request is pending in RPQ and a maintenance operation (MRR, ZQCal, Ref, etc.) needs to be performed, D-unit delays the maintenance operation and allows this many read or write requests to be scheduled before allowing the maintenance operation. Note: This mode does not apply to Panic refreshes.
24:22	0h RO	**Reserved (RSVD24_22):** Reserved
21	0h RW	**Disable Refresh Debt Clear (DISREFDBTCLR):** When set, D-Unit will not clear refresh debt before Self Refresh SR Entry: • 0: D-Unit sends all postponed REF commands to DRAM before it enters Self Refresh. • 1: D-Unit enters SR without clearing the Refresh Debt (for Debug only).

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**Refresh Skew Disable (REFSKWDIS):** Disables Skewing of Refresh Counting between Ranks. Each rank has its own refresh counter. By default incrementing these refresh counters are skewed by 1/2 the tREFI period. Setting this bit to a 1 disables this feature and all refresh counters will increment at the same time per tREFI period. Skewing the tREFI counters can improve performance since traffic to all ranks does not have to be blocked to perform refresh. • 0: Incrementing the refresh counters are skewed by 1/2 tREFI period. • 1: All refresh counters will increment at the same time per tREFI period.
19:18	0h RO	**Reserved (RSVD19_18):** Reserved
17:16	0h RO	**Reserved (RSVD17_16):** Reserved
15	0h RW	**Extra Refresh Debit (EXTRAREFDBT):** When set to 1, D-Unit adds one extra refresh debit (for a total of two) on Self-refresh exit.
14:12	1h RW	**Minimum Refresh Rate (MINREFRATE):** Ensures that refresh rate never drops below a certain limit regardless of TQ polling. • 000: Stop issuing refresh commands and accumulating refresh debits. (does not stop tREFI counter). • 001: 0.25x refresh rate (i.e. 4x tREFI same as no limit). • 010: 0.5x refresh rate (i.e. 2x tREFI). • 011: 1x refresh rate (i.e. 1x tREFI). • 100: 2x refresh rate (i.e. 0.5x tREFI). • 101: 4x refresh rate (i.e. 0.25x tREFI). • 110: 4x refresh rate with derating forced on i.e. 0.25x tREFI. • 111: Reserved.
11:8	7h RW	**Refresh Panic Watermark (REFWMPNC):** When the Refresh counter per rank is greater than this value, the D-Unit will send a REF command to the rank regardless of pending requests. Note: REFWMPNC must be greater than or equal to REFWMHI and greater than 2, Max Value must be less than 8 to not violate 9xtREFI JEDEC requirement.
7:4	5h RW	**Refresh High Watermark (REFWMHI):** When the Refresh counter per rank is greater than this value, the D-Unit will send a REF command to the rank if there is no critical priority requests in the pending queues. Note: Value must be greater or equal to 1 and less than or equal to REFWMPNC.
3:1	0h RO	**Reserved (RSVD3_1):** Reserved
0	0h RW	**Opportunistic Refresh Disable (OREFDIS):** Disable opportunistic scheduling of refresh. • 0: D-Unit will send a REF command only if there is no pending request to that rank. • 1: D-Unit will not send any opportunistic refreshes. Refresh commands are only sent when the refresh counter is greater than REFWMHI. Note: When set, DISREFDBTCLR must also be set to be able to enter SR.

3.57 D-Unit Scheduler Control (D_CR_DSCH) — Offset 123Ch

Specifies parameters to control scheduling of commands to DRAM.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 123Ch	3901C08 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Early Write DQ Enable (EARLY_DQ_EN):** Enables D-unit to send the write data to the PHY one clock earlier than WL value (WL - 1). Note: Applicable to DDR3L and DDR4 only.
30:29	0h RW	**BGF Early Read Data Valid (BGF_EARLY_RDDATA_VALID):** Specifies the number of clocks the D-Unit sends the read data valid through the BGF earlier as compared to the data. • 00: Always write read valid in same SPID clock as data (STATIC_0). • 01: Always write read valid one SPID clock before data (STATIC_1). • 10: Write read valid up to 2 SPID clocks before data (DYNAMIC). • 11: Reserved
28:27	0h RW	**SPID Early Read Data Valid (SPID_EARLY_RDDATA_VALID):** Specifies the delay in SPID clocks from RDDATA_VALID assertion to actual data on SPID. The value should match what is programmed in DDRIO (PHY).
26:21	1Ch RW	**Write Pending Queue Count (WPQCOUNT):** Used to limit the number of available slots in Write Pending Queue/ Write Data Buffer. WPQCOUNT will only recognize changes when PMI ISM is not active.
20:16	10h RW	**Read Pending Queue Count (RPQCOUNT):** Used to limit the number of entries in Read Pending Queue. RPQCOUNT will only recognize changes when PMI ISM is not active.
15	0h RO	**Reserved (RSVD15):** Reserved
14:10	7h RW	**Read Return Data Additional Credits (BLKRDBF_ADD_RDDATA_CR):** Number of additional full cacheline (64B) read data return credits exposed to D-Unit when BLKRDBF is set. Note: The value in this field has no effect on Read return credits when BLKRDBF is not set.
9:8	0h RW	**In-Order Mode (INORDERMODE):** • 0h: In order mode disabled: Commands are sent out of order. • 1h: Partial in order mode: Read and Write CAS commands are sent in the order they were received. ACT and PRE can go out of order. • 2h: Full in order mode serialized test: All DRAM commands CAS ACT PRE associated with a PMI request are issued to DDR before any DRAM commands for a subsequent PMI request. • 3h: Reserved. In order modes should be enabled during init/training/CPGC testing. Should never be changed while the D-Unit queues are nonempty. For WIO2, when in-order mode is enabled (01 or 10), D_CR_DSCH_BYPASSEN must be set to 0
7	0h RW	**Idle Bypass Mode Enable (BYPASSEN):** When set new page hit/empty read requests will bypass D-Unit pipeline stages to save latency 0 - Disable Idle Bypass 1 - Enable Idle Bypass Note: Only applies to WIO2, this bit is reserved for other technologies
6	0h RW	**Block When RDB Full (BLKRDBF):** When set D-Unit stops scheduling new read commands to DRAM when the read data buffer (RDB) is full.
5:4	0h RW	**Stretch Mode (STRETCHMODE):** When stretch mode is enabled, commands are initiated only on Phase 0 of SPIDClk. • 00: Stretch mode is disabled. • 01: Commands are initiated on Phase 0 of every SPID clocks. • 10: Commands are initiated on Phase 0 of even SPID clocks. • 11: Commands are initiated on Phase 0 of odd SPID clocks.
3:0	8h RW	**Masked Write Turnaround Delta (TMWR_TA_DELTA):** The value in this register is subtracted from Masked Write to Read, Masked Write to Write and Masked Write to Masked Write turnaround times to account for half BL MWr commands in LPDDR4 and WIO2. • LPDDR4: = MWr tCCD = MWr BL/2 = 8. • WIO2: NA.

3.58 DRAM Calibration Control (D_CR_DCAL) — Offset 1240h

Specifies parameters to control ZQ Calibration.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1240h	1057 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**ZQ Calibration Type (ZQCALTYPE):** Determines whether the ZQ Calibration is a long or short calibration command (due to ZQCALSTRT). 0: Short calibration (ZQCS). 1: Long calibration (ZQCL).
30	0h RW/V	**ZQ Calibration Start Rank 1 (ZQCALSTRTR1):** Set this bit to 1 to start the ZQ calibration sequence on Rank 1. This bit will remain a 1 until the ZQ calibration is complete for rank 1, then it will return to 0. 0: ZQ calibration is done. 1: ZQ calibration has started and is in progress.
29	0h RW/V	**ZQ Calibration Start Rank 0 (ZQCALSTRTR0):** Set this bit to 1 to start the ZQ calibration sequence on Rank 0. This bit will remain a 1 until the ZQ calibration is complete for rank 0, then it will return to 0. 0: ZQ calibration is done. 1: ZQ calibration has started and is in progress.
28:23	0h RO	**Reserved (RSVD28_23):** Reserved
22:21	0h RW	**Self-Refresh Exit ZQ Calibration Control (SRXZQC):** • 00: On DynSR exit ZQ timer determines the ZQ type. When the state is lost (i.e due to AutoPG/S0ix) ZQCL is always performed. • 01: Always perform ZQCL after self refresh exit. In LPDDR4, ZQ with traffic blocked. • 10: Always perform ZQCS on SR exit. For LPDDR4, ZQ while traffic is allowed. • 11: No ZQCL commands are sent (it disables ZQCAL commands on SR exit).
20:18	0h RO	**Reserved (RSVD20_18):** Reserved
17	0h RW	**ZQ Calibration Mode (ZQCLMODE):** Specifies how ZQCal commands are sent to different ranks. • 0: ZQCal commands are sent in parallel to all ranks. • 1: ZQCal commands are sent serially to each rank.
16	0h RW	**Periodic ZQ Calibration Disable (ZQCDIS):** • 0: Periodic ZQ Calibration is Enabled. • 1: Disable periodic ZQ Calibration.
15:14	0h RO	**Reserved (RSVD15_14):** Reserved
13:0	1057h RW	**ZQ Calibration Interval (ZQINT):** Specifies the time interval between two ZQCS (LPDDR3/DDR3L/DDR4) or ZQ Start (LPDDR4) commands to a DRAM device. (in RTC 32.8KHz clocks)

3.59 DRAM Mode Registers Shadow Copy (D_CR_MR_SHADOW) — Offset 1248h

This register contains a copy of Mode Regsiters in the DRAM so that D-unit can only modify certain bits without affecting other values.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1248h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:16	0h RW	**MR Value 2 (MR_VALUE2):** MR3 Shadow Register (DDR4): BIOS writes the correct value of MR3 register in DDR4 into this field at boot time. D-Unit modifies one bit and rewrites the MR3 into DDR4 to enter and exit MPR mode.
15:14	0h RO	**Reserved (RSVD15_14):** Reserved
13:0	0h RW	**MR Value (MR_VALUE):** MR3 Shadow Register (WIO2): BIOS sets the value of this field at boot time based on the DRAM device configuration. D-Unit merges the value in MR3_THERM_OFFSET with this field and writes the result into DRAM MR3 MR2 Shadow Register (DDR3L): BIOS writes the correct value of MR2 register in DDR3L into this field at boot time. D-Unit modifies one bit and rewrites the MR2 into DDR3L DRAM before SR entry. MR4 Shadow Register (DDR4): BIOS writes the correct value of MR4 register in DDR4 into this field at boot time. D-Unit modifies one bit and rewrites the MR4 into DDR4 DRAM after temperature read out.

3.60 VNN Scaling Timer Control (D_CR_VNNTIMER) — Offset 124Ch

Specifies parameters for VNN Scaling Timer in D-Unit. The values in this register will be set by P-code during VNN scaling period.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 124Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**VNN Scaling Timer Enable (VNN_TIMER_EN):** • 0: The D-Unit VNN Scaling Timer is disabled. • 1: The D-Unit VNN Scaling Timer is enabled.
30:12	0h RO	**Reserved (RSVD30_12):** Reserved
11:0	0h RW	**VNN Timer Time (VNN_TIMER_TIME):** The final timer value (in 16 x DRAM clocks).

3.61 Periodic DRAM Temperature Polling Control (TQ) (D_CR_TQCTL) — Offset 1250h

Specifies the control for periodic temperature monitoring and control of DRAM device.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1250h	6C000008 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	3h RW/V	**TQ Data Rank 1 (TQDATAR1):** If Rank 1 is disabled, this value will remain zero. This field contains the data of the last temperature sensor read from Rank 1 DRAM Mode Register. It is overwritten with each command.
28:26	3h RW/V	**TQ Data Rank 0 (TQDATAR0):** This field contains the data of the last temperature sensor read from Rank 0 DRAM Mode Register. It is overwritten with each command.
25:22	0h RO	**Reserved (RSVD25_22):** Reserved
21:8	0h RW	**TQ Poll Period (TQPOLLPER):** This sets the frequency by which the D-Unit initiates temperature sensor read from DRAM mode register to determine required refresh rate (in 4x tREFI units).
7	0h RW	**Temperature Controlled Refresh Range Enable (DDR4 Only) (TCRREN):** When set, after a DRAM temperature read out (MPR sequence), D-unit writes a 1 to bit 2 of MR_SHADOW.MR_VALUE when temperature sensor read out for that rank indicates a value higher than 0x1, and writes a 0 to that bit otherwise. The new MR_VALUE is then written into MR4 of DDR4 for each enabled rank.
6	0h RW/V	**TQ Poll Start Rank 1 (TQPOLL_START_R1):** When set Dunit will initiate temperature sensor read for Rank 1. Hardware will clear the bit once the MR read command is issued for this rank. Note: TQPOLLEN mus be 0 before this bit is set.
5	0h RW/V	**TQ Poll Start Rank 0 (TQPOLL_START_R0):** When set Dunit will initiate temperature sensor read for Rank 0. Hardware will clear the bit once the MR read command is issued for this rank. Note: TQPOLLEN mus be 0 before this bit is set.
4	0h RW	**Self Refresh Temperature Range Enable (DDR3 Only) (SRTEN):** When set, before every Self refresh entry, D-Unit writes a 1 to bit 7 of MR_SHADOW.MR_VALUE when TQDATA for that rank indicates a value higher then 0x3, and writes a 0 to that bit otherwise. The new MR_VALUE is then written into MR2 of DDR3 for each enabled rank.
3	1h RW	**Enable Dynamic Timing Derating (ENDERATE):** When set to 1, the Dynamic Timing Derating is enabled. When the D-Unit determines (via TQ polling) that the DRAM requires timing derating in addition to refresh interval adjustment, the D-Unit will automatically adjust the relevant timing parameters.
2	0h RW	**Enable TQ Data Push (TQDATAPUSHEN):** When set to 1, D-Unit pushes the data form the last temperature sensor read to a punit register.
1	0h RW	**Enable TQ Poll on Self-Refresh Exit (TQPOLLSREN):** This bit enables temperature sensor read on Self Refresh Exit. If disabled, D-Unit will not initiate MR Read sequence to read temperature value on Self-Refresh exit.
0	0h RW	**Enable Periodic TQ Poll (TQPOLLEN):** This bit enables periodic TQ Poll. If disabled, D-Unit will not read the DRAM's temperature sensor value periodically.

3.62 Temperature Offset Control (D_CR_TQOFFSET) — Offset 1254h

Specifies temperature offset and refresh rate adjustments requested by software.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1254h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD31_16):** Reserved
15:11	0h RO	**Reserved (RSVD15_11):** Reserved
10:8	0h RW	**MR4 Adder (MR4_ADDER):** D-Unit adds the value of this field to TQDATA read from MR4 the resulting value is used to control refresh rate and AC timing derating.
7:3	0h RO	**Reserved (RSVD7_3):** Reserved
2	0h RW/V	**MR3 Offset Update (MR3_OFFSET_UPDATE):** When set, D-Unit writes the merged value of MR3_VALUE and MR3_THERM_OFFSET into MR3 of DRAM. D-Unit clears this bit once the value is written.
1:0	0h RW	**MR3 Thermal Offset (MR3_THERM_OFFSET):** (WIO2 only) Everytime MR3_OFFSET_UPDATE is set, D-Unit merges the value in this field with MR3_VALUE and writes the result to MR3 in DRAM

3.63 D-Unit Control Operations (D_CR_DCO) — Offset 1258h

Specifies D-Unit initialization and control operation.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1258h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/V	**Initialization Complete (IC):** Indicates that initialization of the D-Unit has been completed. Memory accesses are permitted and maintenance operation begins. Until this bit is set to a 1, the memory controller will not accept DRAM requests from the Bunit/GSA/2LM (PMI ISMs will not leave idle). Note: Set this bit to 1 only when all other D-Unit registers have been configured. Usually set at the last configuration step by BIOS on cold/warm reset. D-Unit hardware sets this bit on SR exit.
30	0h RO/V	**DDRIO PHY Initialization Complete (DIOIC):** Status indication that the DDRIO PHY initialization is complete reflects the status spid_init_complete signal.
29	0h RO	**Reserved (RSVD29):** Reserved
28	0h RW	**PMI Control Select (PMICTL):** • 0: D-Unit PMI is connected to Bunit/GSA/2LM. • 1: D-Unit PMI is connected to CPGC. Note: D_CR_DSCH_BYPASSEN must be set to 0 in CPGC mode. Note: PMI must be idle and D-Unit BGF_RUN = 0 before changing the value in this register.
27:8	0h RO	**Reserved (RSVD27_8):** Reserved
7:4	0h RO	**Reserved (RSVD7_4):** Reserved
3	0h RW	**Enable PSMI Mode (PSMIEN):** When enabled, D-Unit will synchronize clock crossing signals. • 0: PSMI Mode is disabled. • 1: PSMI Mode is enabled. Note: Change only allowed when D-Unit is idle.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Maintenance Reset (MNTRST):** Writing a 1 to this field resets all maintenance timers. Clears all states and also clears refresh debt queues. This bit must be cleared by software after at least 3 SPID clocks.
1	0h RW	**Enable Maintenance Operations (MNTEN):** Setting this field to 1 enables all maintenance operations. When DCO.IC is set, the maintenance operations are enabled irrespective of the value of this field.
0	0h RO	**Reserved (RSVD0):** Reserved

3.64 Data Scrambler (D_CR_SCRAMCTRL) — Offset 12A4h

Specifies parameters to control data scrambling in D-Unit.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 12A4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Enable Data Scrambler (SCRM_EN):** When set to 1, data scrambling is enabled. When set to 0, data scrambling is disabled. Should be set before D_CR_BGF_CTL_BGF_RUN is set to 1.
30	0h RO	**Reserved (RSVD30):** Reserved
29:28	0h RW	**Scrambler Clock Gate Select (CLOCKGATE):** This field controls how the scrambler output code is clock gated to reduce power. • 00: Clock gate disabled. • 01: Clock Gate every 2 cycles. • 10: Clock Gate every 3 cycles. • 11: Clock Gate every 4 cycles.
27:16	0h RO	**Reserved (RSVD27_16):** Reserved
15:0	0h RW	**Scrambling Key (KEY):** Sets the key for the scrambler. The key should be a random value that is set following each cold boot.

3.65 Error Injection Address Register (D_CR_ERR_INJ) — Offset 12ACh

Contains the target address for ECC error injection.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 12ACh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD31):** Reserved
30:1	0h RW	**Error Injection Target Address (ADDRESS):** Specifies the PMI address of the write transaction to be injected with the error. Only applicable to Write transactions. Read/ under-fill read of the partial write operation is not affected.
0	0h RO	**Reserved (RSVD0):** Reserved

3.66 Error Injection Control Register (D_CR_ERR_INJ_CTL) — Offset 12B0h

Controls injecting correctable or uncorrectable errors into the write requests specified by target address.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 12B0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RSVD31_4):** Reserved
3	0h RW	**Error Injection Type Higher 32B (SEL_HI):** 0 - Uncorrectable Error (UE) is armed for write address matching to inject UE by using the same poisoning scheme, i.e. inverting corresponding write ECC[6:0] on every QW of the 32B data. 1 - Correctable Error (CE) is armed for write address matching to inject CE by inverting corresponding write ECC[0] on every QW of the 32B data.
2	0h RW	**Error Injection Enable Higher 32B (EN_HI):** When set the error injection is continuously armed for higher 32B of D_CR_ERR_INJ_ADDR write address matching until it is cleared.
1	0h RW	**Error Injection Type Lower 32B (SEL_LO):** 0 - Uncorrectable Error (UE) is armed for write address matching to inject UE by using the same poisoning scheme, i.e. inverting corresponding write ECC[6:0] on every QW of the 32B data. 1 - Correctable Error (CE) is armed for write address matching to inject CE by inverting corresponding write ECC[0] on every QW of the 32B data.
0	0h RW	**Error Injection Enable Lower 32B (EN_LO):** When set, the error injection is continuously armed for lower 32B of D_CR_ERR_INJ_ADDR write address matching until it is cleared.

3.67 Error Log Register (D_CR_ERR_ECC_LOG) — Offset 12B4h

Detected ECC errors are captured in this register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 12B4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/V	**CLEAR:** Setting this bit to one clears all fields in this register, including itself.
30:29	0h RW	**PMI VISA Byte Select (ECC_VISA):** Select ECC or PMI byte on VISA : • 00: ECC byte, • 01: PMI Data Byte [7:0], • 10: PMI Data Byte [63:56], • 11: PMI Data Byte [255:248]
28	0h RW/V	**Correctable Single-bit Error (CERR):** This bit is set when a correctable single-bit error occurs on a memory read data transfer. When this bit is set, the address that caused the error and the error syndrome are also logged and they are locked to further single bit errors, until this bit is cleared. A multiple bit error that occurs after this bit is set will override the address/error syndrome information.
27	0h RW/V	**Uncorrectable Multiple-bit Error (MERR):** This bit is set when an uncorrectable multiple-bit error occurs on a memory read data transfer. When this bit is set, the address that caused the error and the error syndrome are also logged and they are locked until this bit is cleared.
26:25	0h RW/V	**Error Burst Number (ERR_BURST):** Burst number (in BL8) of the error within a chunk.
24	0h RW/V	**Error Chunk Number (ERR_CHUNK):** Chunk number of the error. 0 - lower 32B chunk has error if MERR/CERR is set 1 - higher 32B chunk has the error if MERR/CERR is set
23:16	0h RW/V	**Quad Word ECC Syndrome (SYNDROME_QW):** ECC Syndrome for a QW (64 bit) within 32B Address
15:0	0h RW/V	**Request Tag (TAG):** Read Return Tag matches with the PMI Request Tag which triggered the error log.

3.68 D-Unit Fuse Status (D_CR_DFUSESTAT) — Offset 12BCh

Contains the values read from D-Unit fuses.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 12BCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD31_16):** Reserved
15:0	0h RO/V	**D-Unit Fuse Status (FUSESTAT):** D-Unit fuse bits are captured into this register and are available to be read. • [0]: fus_dun_ecc_dis. • [3:1]: fus_dun_max_supported_device_size[2:0]. • [4:4]: fus_dun_lpddr3_dis. • [5:5]: fus_dun_lpddr4_dis. • [6:6]: fus_dun_wio2_dis. • [7:7]: fus_dun_ddr3l_dis. • [8:8]: fus_dun_ddr4_dis. • [15:9]: reserved.

3.69 Major Mode Control (D_CR_MMC) — Offset 1324h

Specifies parameters to control read/write major mode operation and transitions.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1324h	2B01A518 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:27	5h RW	**RAW Conflict Read Priority for WMM Transition (RAW_WMM):** If a conflict read reaches this priority (or greater depending on access class occupancy), WMM will be triggered to unblock the corresponding write. D-Unit will stay in WMM until corresponding write is issued. Note: The value in this bit must not be higher than lowest terminal priority level of each access class.
26	0h RO	**Reserved (RSVD26):** Reserved
25:23	6h RW	**Read Isoch Trigger Priority (RIMPRIO):** If any read in the RPQ is at this programmable priority, RIM is triggered.
22:18	0h RO	**Reserved (RSVD22_18):** Reserved
17:12	1Ah RW	**Write Isoch Threshold (WIMTHRS):** When the number of entries in WPQ is greater than or equal to this value (higher than WMM entry watermark, less than WPQ size), it triggers write isoch mode (WIM).
11:6	14h RW	**Write Major Mode Exit Watermark (WMMEXIT):** When the number of entries in WPQ is less than this value, the D-Unit will switch back to read major mode.
5:0	18h RW	**Write Major Mode Entry Watermark (WMMENTRY):** When the number of entries in WPQ is greater than or equal to this value, the D-Unit will switch to write major mode (WMM). Note: the value must not be set to 0.

3.70 Major Mode RD/WR Counter (Set A and B) (D_CR_MMRDWR_AB) — Offset 1328h

Minimum read and maximum write counter control. This register defines the minimum number of reads in RMM and maximum number of writes in WMM before a mode transition happens.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1328h	1020408 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (RSVD31_26):** Reserved
25:20	10h RW	**Max Writes B (MAXWRB):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set B).

Bit Range	Default & Access	Field Name (ID): Description
19:14	8h RW	**Min Reads B (MINRDB):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set B).
13:12	0h RO	**Reserved (RSVD13_12):** Reserved
11:6	10h RW	**Max Writes A (MAXWRA):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set A).
5:0	8h RW	**Min Reads A (MINRDA):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set A).

3.71 Major Mode RD/WR Counter (Set C and D) (D_CR_MMRDWR_CD) — Offset 132Ch

Minimum read and maximum write counter control. This register defines the minumum number of reads in RMM and maximum number of writes in WMM before a mode transition happens (sets C and D).

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 132Ch	1020408 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (RSVD31_26):** Reserved
25:20	10h RW	**Max Writes D (MAXWRD):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set D).
19:14	8h RW	**Min Reads D (MINRDD):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set D).
13:12	0h RO	**Reserved (RSVD13_12):** Reserved
11:6	10h RW	**Max Writes C (MAXWRC):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set C).
5:0	8h RW	**Min Reads C (MINRDC):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set C).

3.72 Access Class Initial Priority (D_CR_ACCIP) — Offset 1330h

Each field of this register defines the initial priority of one access class.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1330h	17C2 h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD31_15):** Reserved
14:12	1h RW	**Access Class 4 Initial Priority (AC4IP):** Initial priority level of read requests coming with access class 4.
11:9	3h RW	**Access Class 3 Initial Priority (AC3IP):** Initial priority level of read requests coming with access class 3.
8:6	7h RW	**Access Class 2 Initial Priority (AC2IP):** Initial priority level of read requests coming with access class 2.
5:3	0h RW	**Access Class 1 Initial Priority (AC1IP):** Initial priority level of read requests coming with access class 1.
2:0	2h RW	**Access Class 0 Initial Priority (AC0IP):** Initial priority level of read requests coming with access class 0.

3.73 Access Class 0 Priority Promotion Control (D_CR_RD_PROM0) — Offset 1334h

This register defines the priority promotion policy for access class 0. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1334h	1F52940 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
24:20	1Fh RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	Ah RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	Ah RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	Ah RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.74 Access Class 1 Priority Promotion Control (D_CR_RD_PROM1) — Offset 1338h

This register defines the priority promotion policy for access class 1. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the associated level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1338h	14000000 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	Ah RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
24:20	0h RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	0h RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	0h RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	0h RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.75 Access Class 2 Priority Promotion Control (D_CR_RD_PROM2) — Offset 133Ch

This register defines the priority promotion policy for access class 2. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 133Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

Bit Range	Default & Access	Field Name (ID): Description
24:20	0h RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	0h RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	0h RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	0h RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.76 Access Class 3 Priority Promotion Control (D_CR_RD_PROM3) — Offset 1340h

This register defines the aging policy for access class 3. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1340h	1F29400 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
24:20	1Fh RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	5h RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	5h RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	0h RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.77 Access Class 4 Priority Promotion Control (D_CR_RD_PROM4) — Offset 1344h

This register defines the aging policy for access class 3. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1344h	1F5294A h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
24:20	1Fh RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	Ah RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	Ah RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	Ah RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	Ah RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.78 Deadline Threshold (D_CR_DL_THRS) — Offset 1348h

Specifies when the request with initial priority 0 get promoted to a higher priority level.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1348h	6 h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD31_11):** Reserved
10:0	6h RW	**Deadline Threshold (DEADLINE_THRS):** A requests with initial priority of 0 will exit priority 0 when its deadline is equal or less than this value plus current time. This field does not affect the priority of any requests in access classes with initial priority bigger than 0.

3.79 Major Mode Blocking Rules Control (D_CR_MM_BLK) — Offset 134Ch

This register controls blocking rules enforced in RMM and WMM.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 134Ch	1800 h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD31_25):** Reserved
24	0h RW	**WMM Regular Rule 1 (WMM_REG_R1):** Disable WMM unsafe write page hits block safe write page misses same bank.
23:20	0h RO	**Reserved (RSVD23_20):** Reserved
19	0h RW	**WMM Priority Rule 4 (WMM_PRIO_R4):** Disable WMM unsafe priority 1 read miss block write hit to same bank. Note: This rule does not block the bank that is being blocked by WMM_PRIO_R3. Priority rules 1, 3 and 4 should be enabled/disabled together.
18	0h RW	**WMM Priority Rule 3 (WMM_PRIO_R3):** Disable WMM unsafe priority 1 write hit block write miss to same bank. Note: This rule does not block the bank that is being blocked by WMM_PRIO_R1. Priority rules 1,3 and 4 should be enabled/disabled together.
17	0h RW	**WMM Priority Rule 2 (WMM_PRIO_R2):** Disable WMM CAS block rule.
16	0h RW	**WMM Priority Rule 1 (WMM_PRIO_R1):** Disable WMM unsafe top priority 1 write miss block write hit same bank. Priority rules 1, 3 and 4 should be enabled/disabled together.
15:14	0h RO	**Reserved (RSVD15_14):** Reserved
13	0h RW	**RMM Regular Rule 6 (RMM_REG_R6):** Disable RMM unsafe write page hits block safe write page misses same bank.
12	1h RW	**RMM Regular Rule 5 (RMM_REG_R5):** Disable RMM unsafe read page miss block all safe and unsafe write page hit to the same bank. Note: This field must not be set to 0 (enabled) if RMM_REG_R4 is also 0.
11	1h RW	**RMM Regular Rule 4 (RMM_REG_R4):** Disable RMM unsafe write page hit block safe read page miss same bank. Note: This field must not be set to 0 (enabled) if RMM_REG_R5 is also 0.
10	0h RW	**RMM Regular Rule 3 (RMM_REG_R3):** Disable RMM unsafe read page hit block safe read and write page miss same bank. Note: This rule does not block the bank that is being blocked by RMM_PRIO_R3 and RMM_PRIO_R1.
9	0h RW	**RMM Regular Rule 2 (RMM_REG_R2):** Disable RMM unsafe read page empty block safe write page empty same rank.
8	0h RW	**RMM Regular Rule 1 (RMM_REG_R1):** Disable RMM unsafe read page hit block safe write page hit same rank.
7:4	0h RO	**Reserved (RSVD7_4):** Reserved
3	0h RW	**RMM Priority Rule 4 (RMM_PRIO_R4):** Disable RMM unsafe critical read miss block read and write hit to same bank. Note: This rule does not block the bank that is being blocked by RMM_PRIO_R3. Priority rules 1, 3 and 4 should be enabled/disabled together.
2	0h RW	**RMM Priority Rule 3 (RMM_PRIO_R3):** Disable RMM unsafe critical read hit block read and write miss to same bank. Note: This rule does not block the bank that is being blocked by RMM_PRIO_R1. Priority rules 1, 3 and 4 should be enabled/disabled together.
1	0h RW	**RMM Priority Rule 2 (RMM_PRIO_R2):** Disable RMM CAS block rule.
0	0h RW	**RMM Priority Rule 1 (RMM_PRIO_R1):** Disable RMM unsafe top critical read miss block read and write hit same bank. Note: Priority rules 1, 3 and 4 should be enabled/disabled together.

3.80 DRAM Self-Refresh Command (D_CR_DRAM_SR_CMD) — Offset 1354h

Self refresh command register to allow sending WAKE and SUSPEND messages to D-Unit. (Only one bit can be set at a time). Posted writes to this register are not completed until hardware clears the field.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1354h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RSVD31_4):** Reserved
3	0h RW/V	**SUSPENDP:** A SUSPENDP message will put the DRAM into self-refresh mode. The D-Unit will complete servicing outstanding memory requests and flush all queued Refresh commands to DRAM before putting the DRAM in self refresh mode. Finally, a PM message will be sent to the PHY. The bit is cleared by hardware after the PHY indicates the transition requested in the PM message has been completed. D-Unit will perform an MRW to MR17 with an opcode as defined by DPMC0.PASR before it places the DRAM into Self-Refresh.
2	0h RW/V	**SUSPEND:** A SUSPEND message will put the DRAM into self-refresh mode. The D-Unit will complete servicing outstanding memory requests and flush all queued Refresh commands to DRAM before putting the DRAM in Self Refresh mode. Finally, a PM message will be sent to the PHY. The bit is cleared by hardware only after the PHY indicates the transition requested in the PM message has been completed. Note: When COLDWAKE is set prior of setting this bit the DRAM will not be placed in SR.
1	0h RO	**Reserved (RSVD1):** Reserved
0	0h RW/V	**WAKE:** Take PHY out of PM states and wakes the DRAM out of self refresh mode. The bit is cleared by hardware only when the DRAM has exited out of self refresh mode and is accessible. Note: When COLDWAKE is set prior of setting this bit the D-Unit will not send SR exit command and will not set the DCO.IC bit.

3.81 DQS Retraining Control (D_CR_DQS_RETRAINING_CTL) — Offset 1380h

LPDDR4 DQS Retraining control register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1380h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DQS Periodic Retraining Interval (DQS_RETRAIN_INT):** This sets the frequency by which the D-Unit initiates periodic retraining (in 1x NREFI).
15:14	0h RO	**Reserved (RSVD15_14):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:4	0h RW	**DQS Oscillator Runtime (DQS_OSC_RT):** After D-Unit starts DQS oscillator, it must wait this amount of time before being able to read the value in MR18 and MR19 (in 16x DRAM clocks). Value in this register must be at least equal to DRAM's MR23 value. + tOSCO.
3	0h RW/V	**DQS Retrain Start Rank 1 (DQS_RETRAIN_START_R1):** When set, Dunit will initiate LPDDR4 DQS training sequence on Rank1. Hardware will clear bit when sequence completes (after issuing MR19). Note: DQS_RETRAIN_EN must be 0 before setting this bit.
2	0h RW/V	**DQS Retrain Start Rank 0 (DQS_RETRAIN_START_R0):** When set, Dunit will initiate LPDDR4 DQS training sequence on Rank0. Hardware will clear bit when sequence completes (after issuing MR19). Note: DQS_RETRAIN_EN must be 0 before setting this bit.
1	0h RW	**DQS Retrain SRX Exit (DQS_RETRAIN_SRX_EN):** Enable retraining on SR exit. This bit enables LPDDR4 DQS retraining on Self Refresh Exit. If disabled, D-Unit will not perform retraining on SR exit.
0	0h RW	**DQS Retrain Enable (DQS_RETRAIN_EN):** Periodic retraining enable: This bit enables periodic DQS retraining. If disabled, D-Unit will not perform retraining periodically. Note: Will be enabled only if DCO.IC is set and refreshes are enabled in DRF.MINREFRATE.

3.82 MR4 De-Swizzle Control (D_CR_MR4_DESWIZZLE) — Offset 1384h

Controls the data bits swizzling crossbar for MR4.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1384h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD31):** Reserved
30:28	0h RW	**MR4 Bit 2 Select 2nd Byte (MR4_BIT2_SEL2):** Selects bit 2 of MR4 data.
27	0h RO	**Reserved (RSVD27):** Reserved
26:24	0h RW	**MR4 Bit 1 Select 2nd Byte (MR4_BIT1_SEL2):** Selects bit 1 of MR4 data
23	0h RO	**Reserved (RSVD23):** Reserved
22:20	0h RW	**MR4 Bit 0 Select 2nd Byte (MR4_BIT0_SEL2):** Selects bit 0 of MR4 data.
19:18	0h RO	**Reserved (RSVD19_18):** Reserved
17:16	0h RW	**MR4 Byte 2 Select (MR4_BYTE_SEL2):** Selects byte position of the MR4 data for second device.
15	0h RO	**Reserved (RSVD15):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
14:12	0h RW	**MR4 Bit 2 Select (MR4_BIT2_SEL):** Selects bit 2 of MR4 data.
11	0h RO	**Reserved (RSVD11):** Reserved
10:8	0h RW	**MR4 Bit 1 Select (MR4_BIT1_SEL):** Selects bit 1 of MR4 data.
7	0h RO	**Reserved (RSVD7):** Reserved
6:4	0h RW	**MR4 Bit 0 Select (MR4_BIT0_SEL):** Selects bit 0 of MR4 data.
3:2	0h RO	**Reserved (RSVD3_2):** Reserved
1:0	0h RW	**MR4 Byte Select (MR4_BYTE_SEL):** Selects byte position of the MR4 data first device.

3.83 DRAM Rank Population 0 (D_CR_DRP0) — Offset 1400h

Rank configuration register.

This register is start of dunit 10 LPRRD4 Channel 0, DDR4 Channel 0.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1400h	10000000 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RW	**DRAM Device Per Rank (DRAMDEVICE_PR):** Specifies the number of DRAM devices that are ganged together to form a single rank. • 00: 1 DRAM device in each rank. • 01: 2 DRAM devices in each rank. • 10: 4 DRAM devices in each rank. • 11: 8 DRAM devices in each rank. Note: The actual number of devices is one more than the value programmed when ECC is enabled.
29:28	1h RW	**Address Decode (ADDRDEC):** Specifies the address mapping to be used: • 00: 1KB (A). • 01: 2KB (B). • 10: 4KB (C). • 11: Reserved.
27:25	0h RW	**Burst Length Mode (BLMODE):** • 000: Fixed BL8. • 001: Onthefly BL8. • 010: Fixed BL16. • 011: Onthefly BL16. • 100: Fixed BL32. • 101: Onthefly BL32. • 110-111: Reserved.

Bit Range	Default & Access	Field Name (ID): Description
24:22	0h RW	**DRAM Type (DRAMTYPE):** • 000: Reserved. • 001: Reserved. • 010: LPDDR4. • 011: Reserved. • 100: DDR4. • 101-111: Reserved. Note: The D-Unit should only use this field if allowed by fuse.
21	0h RW	**ECC Enable (ECCEN):** • 0: ECC is disabled. • 1: ECC is enabled. This bit determines if the D-Unit treats the PMI BE_ECC bits as ECC bits or Byte Enables. The D-Unit should not allow this bit to be set if ECC is disabled by fuse. This should only be used in configurations that support ECC.
20:19	0h RW	**CA Swizzle Type (CASWIZZLE):** • 00: uniDIMM/SODIMM/UDIMM. • 01: BGA. • 10: NA. • 11: uniDIMM/SODIMM/UDIMM with Rank 1 mirrored (DDR4 Only).
18:17	0h RO	**Reserved (RSVD18_17):** Reserved
16	0h RW	**Fine Bank Group Interleaving Enable (FBGINTEN):** When enabled, D-unit will use a lower order address bit in the bank hashing function (DDR4 Only). • 0: Bank Group Interleave disabled. • 1: Bank Group Interleave enabled. Note: BAHEN must be set for this bit to take effect.
15	0h RW	**Bank Address Hashing Enable (BAHEN):** • 0: Bank Address Hashing disabled. • 1: Bank Address Hashing enabled.
14	0h RW	**Rank Select Interleave Enable (RSIEN):** • 0: Rank Select Interleaving disabled. • 1: Rank Select Interleaving enabled.
13:9	0h RO	**Reserved (RSVD13_9):** Reserved
8:6	0h RW	**DRAM Device Density (DDEN):** Density of the DRAM devices populated on Ranks 0 and 1. • 000: 4 Gb. • 001: 6 Gb. • 010: 8 Gb. • 011: 12 Gb. • 100: 16 Gb. • 101-111: Reserved. Note: For LPDDR4 this value is the die density.
5:4	0h RW	**DRAM Device Data Width (DWID):** Data width of the DRAM device populated on Ranks 0 and 1. • 00: x8. • 01: x16. • 10: x32. • 11: x64.
3	0h RO	**Reserved (RSVD3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Dual Data Mode Enable (DDMEN):** • 0: PMI Dual Data Mode is disabled in D-Unit. • 1: PMI Dual Data Mode is enabled. Note: Dual Data Mode must be enable for DDR3L/DDR4 configurations.
1	0h RW	**Rank Enable 1 (RKEN1):** Enable Rank 1: Must be set to 1 to enable use of this rank.
0	0h RW	**Rank Enable 0 (RKEN0):** Enable Rank 0: Must be set to 1 to enable use of this rank. Note: Setting this bit to 0 is not a functional mode.

3.84 DRAM Timing Register 9A (D_CR_DTR9A) — Offset 1404h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1404h	14A546 h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RO	**Reserved (RSVD31_23):** Reserved
22:18	5h RW	**Write to Write DQ Delay Same Bank Group Same Rank (TWWSR_L):** Specifies the delay from a DRAM Write to another Write command within the same bank group of the same rank (in DRAM clocks). DDR4 Equation: tWWSR_L = tCCD_L.
17:13	5h RW	**Read to Read DQ Delay Same Bank Group Same Rank (TRRSR_L):** Specifies the delay from a DRAM Read to another Read command within the same bank group of the same rank (in DRAM clocks). DDR4 Equation: tRRSR_L = tCCD_L.
12:6	15h RW	**Write to Read DQ Delay Same Bank Group Same Rank (TWRSR_L):** Specifies the delay from a DRAM Read to Write command within the same bank group of the same rank (in DRAM clocks). • DDR4 Equation: tWRSR_L = CWL + tDQSSmax + BL/2 + tWPST + tWTR_L.
5:0	6h RW	**Row Activation to Row Activation to Same Bank Group Delay [tRRD_L] (TRRD_L):** Specifies the minimum delay (in DRAM clocks) between two DRAM Activate commands to different banks within the same bank group of a rank. (DDR4 Only)

3.85 DRAM Timing Register 0A (D_CR_DTR0A) — Offset 1408h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1408h	820702CB h

Bit Range	Default & Access	Field Name (ID): Description
31:27	10h RW	**Valid Clocks Before CKE High [tCKCKEH/tCSCKEH/tCKSRX] (TCKCKEH):** Number of valid clocks before CKE high (in DRAM clocks). • LPDDR4: The value in this register covers both tCKCKEH and tCSCKEH. • DDR3L/DDR4/LPDDR3/WIO2: The value covers tCKSRX which is defined as the number of valid DRAM clocks that have to toggle before the issuing of the Self Refresh Exit SRX. This value is also used if the clock frequency is changed or the clock is stopped or tristated during Power Down i.e. the number valid DRAM clocks that have to toggle before the issuing of the Power Down Exit PDX command. TCKCKEH can be used to compensate for clock stabilization delays in the motherboard. Note: D-unit hardware enforces minimum of two SPID clock before CKEH, any value in this register is the additional time.
26:22	8h RW	**Exit Self-Refresh to Valid Commands Requiring a Locked DLL Delay [tXSDLL] (TXSDLL):** D-Unit waits max(tXSR+tZQCL/tZQCS, tXSDLL) before allowing traffic to DRAM (in 32 x DRAM Clocks). LPDDR3/LPDDR4/WIO2: tXSDLL = 0. DDR3L/DDR4: tXSDLL = tDLLK. Note: In the equation above, tZQCL/tZQCS = 0 if no ZQ is performed on SR exit.
21:12	70h RW	**Exit Self-Refresh to Valid Command Delay [tXS/tXSR] (TXSR):** DDR3L/DDR4: tXS - Delay between Self Refresh Exit SRX to any DRAM Command not requiring DLL Lock. LPDDR/WIO2: tXSR - Delay between Self Refresh Exit SRX to any DRAM Command. (in DRAM clocks).
11:6	Bh RW	**Activate RAS to CAS Command Delay [tRCD] (TRCD):** Specifies the delay between a DRAM Activate command and a DRAM Read or Write command to the same bank (in DRAM clocks). Note: Derating adds 1.875ns to this timing.
5:0	Bh RW	**Precharge to Activate Command Delay of a Single Bank [tRPpb] (TRPPB):** Specifies the delay between a DRAM Precharge command and a DRAM Activate command to the same bank (in DRAM Clocks). Note : this CR should be constrained to a minimum of 4 in LPDDR3/DDR3L/WIO2 and minimum of 8 in LPDDR4/DDR4. Note: Derating adds 1.875ns to this timing.

3.86 DRAM Timing Register 1A (D_CR_DTR1A) — Offset 140Ch

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 140Ch	30481218 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	6h RW	**Exit Power Down to Next Command Delay [tXP] (TXP):** Specifies the delay from the DRAM Power Down Exit (PDX) command to any valid command (in DRAM clocks). Note: The value in this field must be programmed to tXPDLL when Slow Exit Mode Power-down is enabled for DDR3L.
26	0h RO	**Reserved (RSVD26):** Reserved
25:14	120h RW	**ZQ (long) Calibration Time [tZQCL/tZQCAL] (TZQCL):** • LPDDR3/DDR3L/DDR4: tZQCL/tZQoper - Specifies the delay between the DRAM ZQ Calibration Long (ZQCL) command and any DRAM command during normal operation. • LPDDR4: tZQCAL - ZQ Calibration time (in DRAM clocks). Note: This field defines the ZQ Calibration Long delay during normal operation. It is not the same as tZQinit which uses the same ZQCL command but the delay is longer. tZQinit applies only during poweron initialization of the DRAM devices and tZQoper applies during normal operation. BIOS executes the DRAM initialization sequence so it has to ensure tZQinit is met and not the D-Unit.

Bit Range	Default & Access	Field Name (ID): Description
13:6	48h RW	**ZQ Short Calibration Time [tZQCS] (TZQCS):** ZQCS to any DRAM Command Delay: Specifies the delay between the DRAM ZQ Calibration Short (ZQCS) command and any DRAM command (in DRAM clocks). DDR3L/DDR4/LPDDR3 only. LPDDR4 does not support ZQCS command
5:0	18h RW	**ZQ Latch Time [tZQLAT] (TZQLAT):** Specifies the delay between the DRAM ZQ Calibration Latch command and any DRAM command (in DRAM clocks). LPDDR4 only. Not used in DDR3L/DDR4/LPDDR3/WIO2.

3.87 DRAM Timing Register 2A (D_CR_DTR2A) — Offset 1410h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1410h	46080C30 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	118h RW	**All Bank Refresh Cycle Time [tRFCab] (NRFCAB):** Specifies the delay between the REFab command to the next valid command. (in DRAM clocks)
21	0h RO	**Reserved (RSVD21):** Reserved
20:17	4h RW	**CKE Minimum Pulse Width [tCKE] (TCKE):** Specifies the minimum time from CKEL to CKEH (in DRAM clocks).
16	0h RO	**Reserved (RSVD16):** Reserved
15:0	C30h RW	**Refresh Interval Time [tREFI] (NREFI):** Specifies the average time between refresh commands. JEDEC Base Refresh Interval time (in DRAM clocks). Note: D-Unit will ignore the 2 LSBs of this field.

3.88 DRAM Timing Register 3A (D_CR_DTR3A) — Offset 1414h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1414h	3002EA28 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	6h RW	**Read to Precharge Delay [tRDPRE] (TRTP):** Specifies the minimum delay between the DRAM Read and Precharge commands to the same bank (in DRAM clocks). • LPDDR3 Equation: = BL/2 + tRTP - 4. • LPDDR4 Equation: = BL/2 + Max (8, tRTP) - 8. • WIO2 Equation: NA. • DDR3L/DDR4 Equation: = tRTP.

Bit Range	Default & Access	Field Name (ID): Description
26:20	0h RW	**CAS to CAS Command Delay Adder (TCCD_INC):** Specifies the number of clocks to be added to turnaround times (for Stretch Mode). It increases delay between Read to Read or Read to Write commands by the amount programmed in this field (in 4 x DRAM clocks).
19:13	17h RW	**Write to Precharge Command Delay [tWRPRE] (TWTP):** Specifies the minimum delay between the DRAM Write command and the Precharge command to the same bank (in DRAM clocks). • LPDDR3/LPDDR4/WIO2 Equation: tWTP = BL/2 + WL + tWR + 1. • DDR3L/DDR4 Equation: tWTP = BL/2 + CWL + tWR.
12:11	1h RW	**DRAM Command Valid Duration (TCMD):** Specifies the number of DRAM clocks a command is held valid on the DRAM Address and Control buses. 1N is the DDR3 basic requirement. 2N is the extended mode for board signal integrity. • 0h: Reserved. • 1h: 1 DRAM Clock (1N). • 2h: 2 DRAM Clocks (2N). • 3h: Reserved. Note: DDR3L/DDR4 only. tCMD must be set to 1N for LPDDR3/LPDDR4/WIO2.
10:6	8h RW	**Write Latency [WL/CWL] (TCWL):** The delay between the internal write command and the availability of the first word of DRAM input data (in DRAM clocks).
5:0	28h RW	**Write CAS to Masked Write CAS Delay Same Bank (TWMWSB):** Specifies the minimum delay between DRAM Write command to Masked Write command to same bank (in DRAM clocks). • LPDDR4 Equation: tWMWSB = tCCDMW (BL16) or tCCDMW + 8 (BL32). • WIO2 Equation: NA. Note: Masked Write operation in LPDDR4 is always BL16 and in WIO2 is always BL4. D-Unit applies this timing for same rank as well as same bank, refer to D-Unit for more details.

3.89 DRAM Timing Register 4A (D_CR_DTR4A) — Offset 1418h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1418h	30209149 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	30h RW	**Four Bank Activate Window [tFAW] (TFAW):** A rolling timeframe in which a maximum of four Activate commands can be issued to the same rank. This is to limit the peak current draw from the DRAM devices (in DRAM clocks).
23:18	8h RW	**Write to Read DQ Delay Different Ranks (TWRDR):** Specifies the delay from the start of a Write data burst of one rank to the start of a Read data burst of a different rank (in DRAM clocks). • LPDDR3 Equation: tWRDR = WL + tDQSSmax + BL/2 - (RL + tDQSCKmin - tRPRE). • LPDDR4 Equation: tWRDR = WL - RL + BL/2 + 4 - tDQSCKmin. • DDR3L/DDR4 Equation: tWRDR = CWL + tDQSSmax + BL/2 - (CL + tDQSCKmin - tRPRE). Note: For LPDDR3/4 using ODT, this latency may need to be increased by tODToffadj.

Bit Range	Default & Access	Field Name (ID): Description
17:12	9h RW	**Read to Write DQ Delay Different Ranks (TRWDR):** Specifies the delay from the start of a Read data burst of one rank to the Start of a Write data burst of a different rank (in DRAM clocks). • LPDDR3/LPDDR4 Equation: tRWDR = RL + tDQSCKmax + BL/2 - (WL - tWPRE). • DDR3L/DDR4 Equation: tRWDR = CL + tDQSCKmax + BL/2 - (CWL - tWPRE). Note: For LPDDR3/4 using ODT, this latency may need to be adjusted by tODTon. Note: For DDR3L/DDR4 using ODT, this latency may need to be increased by one clock.
11:6	5h RW	**Write to Write DQ Delay Different Ranks (TWWDR):** Specifies the delay from the start of a Write data burst of one rank to the start of a Write data burst of a different rank (in DRAM clocks). • LPDDR3/DDR3L/DDR4 Equation: tWWDR = BL/2 + tDQSSmax - tDQSSmin + tWPRE. • LPDDR4 Equation: tWWDR = BL/2 + 4 - tDQSSmin. Note: For LPDDR3/4 using ODT, this latency may need to be increased by tODToffadj.
5:0	9h RW	**Read to Read DQ Delay Different Ranks (TRRDR):** Specifies the delay from the start of a Read data burst of one rank to the Start of a Read data burst of a different rank (in DRAM clocks). • Equation: tRRDR = BL/2 + tDQSCKmax - tDQSCKmin + tRPRE.

3.90 DRAM Timing Register 5A (D_CR_DTR5A) — Offset 141Ch

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 141Ch	304200C2 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	6h RW	**Row Activation to Row Activation Delay [tRRD] (TRRD):** Specifies the minimum delay in DRAM clocks between two DRAM Activate commands to the same rank but different banks (tRC is the minimum delay between activations of the same bank). Note: Derating adds 1.875ns to this timing.
26	0h RO	**Reserved (RSVD26):** Reserved
25:23	0h RW	**Derate Increment (TDERATE_INC):** Specifies the additional delay that is added to DRAM timing when indicated by MR4 status. (in DRAM clocks) LPDDR3/LPDDR4/WIO2: Value is 1.875ns. Note: The value in this register is only added to these timing parameters: tRCD, tRAS, tRP and tRRD.
22:18	10h RW	**Write to Write DQ Delay Same Rank (TWWSR):** Specifies the delay from a DRAM Write to another Write command of the same rank (in DRAM clocks). Equation: tWWSR = tCCD.
17:13	10h RW	**Read to Read DQ Delay Same Rank (TRRSR):** Specifies the delay from a DRAM Read to another Read command of the same rank (in DRAM clocks). Equation: tRRSR = tCCD.
12:6	3h RW	**Write to Read DQ Delay Same Rank (TWRSR):** Specifies the delay from a DRAM Read to Write command of the same rank (in DRAM clocks). • LPDDR3/LPDDR4/WIO2 Equation: tWRSR = WL + tDQSSmax + BL/2 + tWTR. • DDR3L Equation: tWRSR = CWL + tDQSSmax + BL/2 + tWTR. • DDR4 Equation: tWRSR = CWL + tDQSSmax + BL/2 + tWTR_S.

Bit Range	Default & Access	Field Name (ID): Description
5:0	2h RW	**Read to Write DQ Delay Same Rank (TRWSR):** Specifies the delay from a DRAM Read to a Write command of the same rank (in DRAM clocks). • LPDDR3/LPDDR4/WIO2 Equation: tRWSR = RL + tDQSCKmax + BL/2 - WL + tWPRE. • DDR3L/DDR4 Equation: tRWSR = CL + tDQSCKmax + BL/2 - CWL + tWPRE. Note: For LPDDR3/4 using ODT, this latency may need to be increased by tODToffadj. Note: For DDR3L/DDR4 using ODT, this latency may need to be increased by one clock.

3.91 DRAM Timing Register 6A (D_CR_DTR6A) — Offset 1420h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1420h	20100000 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	20h RW	**Opportunistic Refresh Idle Timer (OREFDLY):** Rank idle period that defines an opportunity for refresh (in DRAM clocks).
23:19	2h RW	**Valid Clocks After CKE Low [tCKELCK/tCKELCS/tCPDED/tCKSRE] (TCKCKEL):** Specifies the amount of time that DRAM clocks need to toggle after CKE goes low (in DRAM Clocks). • For WIO2/LPDDR3, this covers tCPDED. • For LPDDR4, this covers both tCKELCK and tCKELCS. • For DDR3L/DDR4, this is tCKSRE. Note: D-Unit hardware enforces minimum of one SPID clocks after CKEL, any value in this register is the additional time.
18:15	0h RW	**Reserved (RSVD18_15):** Reserved
14:8	0h RW	**Mode Register Read to Any Command Delay (TPSTMRRBLK):** Specifies the quiet time after issuing MRR command (in DRAM Clocks). This is the channel block time for the MR19 command issued as part of LPDDR4 DQS Retraining flow. For all other MR commands this is the rank block time. Note: D-Unit treats MRR as a read and always applies relevant turnaround times, any value programmed in this CR must be greater than those turnaround times for D-Unit to enforce any additional time from MRR to the next read/write.
7	0h RO	**Reserved (RSVD7):** Reserved
6:0	0h RW	**Any Command to Mode Register Read/Write Delay (TPREMRBLK):** Specifies the channel quiet time before issuing MRR/MRW command. (in DRAM clocks). • LPDDR3 Equation: = tXP + tMRRI - 4. • LPDDR4 Equation: = tXP + tMRRI - 8. • DDR3L Equation: = CWL + BL/2 + 1 - 8 • DDR4 Equation: = CWL + BL/2 + 1 - 16. Note: D-Unit treats MRR as a read and MRW/MRS as a write and always applies relevant turnaround times, any value programmed in this CR must be greater than those turnaround times for D-Unit to enforce any additional time from previous read/writes.

3.92 DRAM Timing Register 7A (D_CR_DTR7A) — Offset 1424h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1424h	D060C06 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	3h RW	**All Bank Precharge to Activate Command Delay [tRPab] (TRPAB):** Specifies the delay between a DRAM Precharge All Bank command and a DRAM Activate command (in DRAM Clocks). Note: This CR should be constrained to a minimum of 4 in LP3 and minimum of 8 in LP4. Note: Derating adds 1.875ns to this timing. • For LPDDR, tRPpb = tRP, tRPab = tRP + 3ns. • For DDR3L, DDR4 and WIO2 8ch tRPpb = tRPab = tRP.
25:23	2h RW	**Mode Register Write to any Command Delay [tMRD/tMRW] (TPSTMRWBLK):** Specifies the quiet time after issuing MRW command (in 8 x DRAM clocks). Note: This time covers for both tMRD and tMRW.
22:16	6h RW	**Write Command to Power Down Delay [tWRPDEN] (TWRPDEN):** Specifies the minimum time between a write command to PowerDown command (in DRAM clocks). Must be at least equal to tWR + tCCD + tWL + 2.
15:9	6h RW	**Read Command to Power Down Delay [tRDPDEN] (TRDPDEN):** Specifies the minimum time between a read command to PowerDown command (in DRAM clocks). Must be at least equal to CL/RL + tDQSCKmax + tCCD + 1.
8:7	0h RO	**Reserved (RSVD8_7):** Reserved
6:0	6h RW	**Row Activation Period [tRAS] (TRAS):** Specifies the minimum delay between the DRAM Activate and Precharge commands to the same bank (in DRAM clocks). Note: Derating adds 1.875ns to this timing.

3.93 DRAM Timing Register 8A (D_CR_DTR8A) — Offset 1428h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1428h	CC50A18 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	3h RW	**Minimum Self-Refresh Time [tSR/tCKESR] (TCKESR):** Specifies the minimum time that DRAM should remain in SR (in DRAM clocks).
25:21	6h RW	**Minimum Low Power Mode Residency (LPMDRES):** Specifies the minimum time that PHY should remain in LPMode (in DRAM clocks).
20:15	Ah RW	**Low Power Mode Exit to Clock Enable Delay (LPMDTOCKEDLY):** Specifies the minimum time between the LP Mode exit to the CK stop/tristate deassertion and powerdown exit (in DRAM clocks). Note: Must be equal to t_idle_latency published in the DDRIO PHY and less than 0x3C.
14:8	Ah RW	**Clock Stop to Low Power Mode Delay (CKETOLPMDDLY):** Specifies the time between CK stop/tristate to the Low Power Mode entry. This timing parameter is used to delay Low Power Mode entry (in DRAM clocks). Note: Must be at least equal to t_idle_length parameter published in the DDRIO PHY and less than 0x7C.
7:0	18h RW	**Power Down Idle Timer (PWDDLY):** This is a non-JEDEC timing parameter used to delay powerdown entry (in DRAM clocks).

3.94 D-Unit ODT Control Register A (D_CR_DOCRA) — Offset 142Ch

Specifies the parameters to control DRAM ODT.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 142Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29	0h RW	**Rank 1 Read ODT Control (R1RDODTCTL):** Specifies the behavior of ODT signals when a Read command is issued to Rank 1. 0 - Read ODT is disabled for Rank 1 1 - Assert ODT to for Rank 0 (non-targeted Rank) Note: This register should be set to 0 for LPDDR3 devices
28	0h RW	**Rank 0 Read ODT Control (R0RDODTCTL):** Specifies the behavior of ODT signals when a Read command is issued to Rank 0. 0 - Read ODT is disabled for Rank 0 1 - Assert ODT to for Rank 1 (non-targeted Rank) Note: This register is reserved for LPDDR3 devices
27:26	0h RW	**Rank 1 Write ODT Control (R1WRODTCTL):** Specifies the behavior of ODT signals when a Write command is issued to Rank 1. 00 - Write ODT is disabled 01 - Assert ODT to Rank 0 (non-targeted Rank) 10 - Assert ODT to Rank 1 (targeted Rank) 11 - Assert ODT to Rank 0 and Rank 1 Note: 10 and 11 are reserved values for LPDDR3
25:24	0h RW	**Rank 0 Write ODT Control (R0WRODTCTL):** Specifies the behavior of ODT signals when a Write command is issued to Rank 0. 00 - Write ODT is disabled 01 - Assert ODT to Rank 0 (targeted Rank) 10 - Assert ODT to Rank 1 (non-targeted Rank) 11 - Assert ODT to Rank 0 and Rank 1 Note: 10 and 11 are reserved values for LPDDR3
23:18	0h RO	**Reserved (RSVD23_18):** Reserved
17:14	0h RW	**Read ODT assertion to de-assertion delay (DDR3L/DDR4) (RDODTSTOP):** Specifies Read ODT assertion to ODT de-assert delay (in DRAM clocks). DDR3L Equation: RDODTSTOP = 6. DDR4 Equation: RDODTSTOP = 5 + tRPRE. Note: Add 1 if DOCRx.RDODTSTART = CL - CWL in 2N mode.
13	0h RO	**Reserved (RSVD13):** Reserved
12:9	0h RW	**Read command to ODT assertion delay (DDR3L/DDR4) (RDODTSTART):** Specifies Read ODT assertion delay after Read Command (in DRAM clocks). DDR3L/DDR4 Equation: RDODTSTART = CL - CWL + tWPRE - tRPRE. Note: In DDR3L/DDR4 2N mode add 1 to enable termination at the start of read data burst.
8:5	0h RW	**Write ODT Assertion to De-assertion Delay (WRODTSTOP):** Specifies number of clocks after ODT assertion that D-Unit deasserts ODT signal (in DRAM clocks). LPDDR3 Equation: WRODTSTOP = RU(tODTon(max)/tCK) + RU((tDQSSmax+tWPST)/tCK) + BL/2 - RD(tODToffmin/tCK) DDR3L Equation: WRODTSTOP = 6. DDR4 Equation: WRODTSTOP = 5 + tWPRE. Note: Add 1 if DOCRx.WRODTSTART = 0 in 2N mode.
4	0h RO	**Reserved (RSVD4):** Reserved
3:0	0h RW	**Write command to ODT assertion delay (WRODTSTART):** Specifies number of clocks after Write command that D-Unit asserts ODT signal (in DRAM clocks). LPDDR3 Equation: WRODTSTART = WL - RU(tODTon(max)/tCK) DDR3L/DDR4 Equation: WRODTSTART = 0 Note: In DDR3L/DDR4 2N mode the value can be set to 0 to assert ODT one DRAM clock earlier than the Write Command (WR) or set to 1 to assert at the same clock as command (CS assertion).

3.95 D-Unit Power Management Control 0 (D_CR_DPMC0) — Offset 1430h

Specifies the parameters to control D-Unit power management features.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1430h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**Reserved (RSVD31_29):** Reserved
28:24	0h RW	**SUSPEND/SUSPENDP Power Management Message Opcode (SUSPMOP):** DDRIO PHY Power Mode Opcode: After the D-Unit has placed the DRAM devices in Self Refresh/PASR mode as the result of a SUSPEND/SUSPENDP message, it sends this 5-bit value to the DDRIO PHY to tell it which power saving mode it should enter. Changing this register value while in SUSPEND will have no effect. Note: This opcode cannot be a PM state where it disables PHY PLLs i.e PM7 in LPDDR PHY.
23	0h RO	**Reserved (RSVD23):** Reserved
22	0h RW	**PM Message Wait for Clock Gate Enable (SRPMCLKW):** Specifies when it is safe to send PM message to the PHY. When enabled, D-Unit waits for SPID Clock to deassert before sending a PM message on SR entry. • 0: D-Unit will not wait for SPID_clk to deassert before sending the PM message to PHY. • 1: D-Unit will wait for SPID_clk to deassert before sending PM message to the PHY. Note: The value must be 1 when DYNPMOP = 7h.
21:17	0h RW	**Dynamic Self-Refresh Power Management Message Opcode (DYNPMOP):** DDRIO PHY Power Mode Opcode: After the D-Unit has placed the DRAM devices in Self Refresh mode as the result of a Dynamic Self-Refresh, it sends this 5bit value to the DDRIO PHY to tell it which power saving mode it should enter. Changing this register value while in self-refresh will only change the PM state for the next entry in DynSR.
16	0h RW	**Dynamic Self-Refresh Enable (DYNSREN):** When set to 1, the D-Unit will automatically control DRAM Self Refresh entry and exit based on interface state and requests in pending queues. When there is no pending request in the queues and PMI is idle, then the D-Unit will place the DRAM devices in Self Refresh mode. The DRAM devices will be brought out of Self-Refresh when idle conditions don't hold.
15:0	0h RW	**Self-Refresh Entry Delay (SREDLY):** Specifies the minimum time the D-Unit will wait before it enters Dynamic Self-Refresh mode when idle (in 16x DRAM Clocks). Note: The value in this field needs to be minimum of 4 in functional mode and minimum of 50 in PSMI mode.

3.96 D-Unit Power Management Control 1 (D_CR_DPMC1) — Offset 1434h

Specifies the parameters to control D-Unit power management features.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1434h	10000028 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29	0h RW	**D-Unit Repeaters Clock Gate Disable (RPTCLKGTDIS):** Setting this bit to 0 allows majority of the repeaters between D-Unit and PHY to clock gate when there is no activity in order to save power. • 0 - Enable Repeaters clock gating. • 1 - Disable Repeaters clock gating. Note: This is a de-feature bit and should be set to 0 for normal operation. Note: The value should only change after DRAM Timing Registers (DTR) are programmed.
28	1h RW	**IOSF-SB End Point Clock Gate Disable (SBEPCLKGTDIS):** Setting this bit to 0 enables the clock gating of IOSF-SB End Points in D-Unit and CPGC when there is no IOSF-SB activity in order to save power. • 0 - Enable IOSF-SB EP clock gating. • 1 - Disable IOSF-SB clock gating. Note: This is a de-feature bit and should be set to 0 for normal operation. Note: The value should only change after DRAM Timing Registers (DTR) are programmed.
27	0h RW	**Local Clock Gate Disable (CLKGTDIS):** Setting this bit to 0 allows the majority of the D-Unit clocks to be gated off when there is no activity in order to save power. When set to 1, D-Unit clockgating is disabled. • 0: Enable. • 1: Disable. Note: This is a de-feature bit and should be set to 0 for normal operation. Note: The value should only change after DRAM Timing Registers (DTR) are programmed.
26	0h RW	**Chip Select Tristate Enable (CSTRIST):** • 0: The DRAM CS pins associated with the enabled ranks are never tristated. • 1: The DRAM CS pins are tristated when DRAM clock is stopped or tristated. Note: CS is not tristated when global tristate flow is disabled (DCBR.TRISTDIS = 1).
25:24	0h RW	**Command/Address Tristate (CMDTRIST):** • 00: The DRAM CA pins are never tristated. • 01: The DRAM CA pins are only tristated when all enabled CKE pins are low. • 10: The DRAM CA pins are tristated when not driving a valid command. • 11: Reserved
23:16	0h RW	**Partial Array Self-Refresh Segment Mask (PASR):** This is the Segment Mask used for the MRW to enable PASR during SUSPENDP (Partial Array Self Refresh entry).
15:8	0h RW	**Page Close Timeout Period (PCLSTO):** Specifies the time from the last access of a DRAM page until that page is scheduled to close by sending a Precharge command to DRAM (in 16 x DRAM clocks).
7	0h RW	**Page Close Timeout Disable (PCLSTODIS):** When disabled, D-Unit will not close the DRAM page when idle. • 0: Enable page close timer. • 1: Disable page close timer (Used during DRAM init and DDRIO training).
6	0h RO	**Reserved (RSVD6):** Reserved
5	1h RW	**ODT Tristate Enable (ODTTRIST):** • 0: The DRAM ODT pins associated with the enabled ranks are never tristated. • 1: DRAMs ODT pins are tristated when DRAM clock is stopped or tristated. Note: ODT is not tristated when global tristate flow is disabled (DCBR.TRISTDIS = 1)

Bit Range	Default & Access	Field Name (ID): Description
4:3	1h RW	**Clock Stop/Tristate Enable (ENCKSTP):** Enable/Disable CK Stop/Tristate During Power down. • 00: Disable CK Stop/Tristate During Power down. • 01: Enable CK Stop During Power down. • 10: Enable CK Tristate During Power down. • 11: Reserved Note: CK is not stopped or tristated when global tristate flow is disabled (DCBR.TRISTDIS = 1).
2:1	0h RW	**Low Power Mode Opcode (LPMODEOP):** D-Unit will send the value in this register after it has entered Powerdown Mode and has stopped/tristated the clock. 00: Disable LPMode. Note: LPMODE entry is not possible when global tristate flow is disabled (DCBR.TRISTDIS = 1).
0	0h RW	**Disable Power Down (DISPWRDN):** Setting this bit to 1 disables dynamic control of DRAM Power-Down entry and exit by keeping the CKE pins driven high. BIOS may set it to 1 during DRAM initialization and DDRIO training. This bit should be set to 0 for normal operation. • 0: The D-Unit dynamically controls the CKE pins to place the DRAM devices in Power Down mode and bring them out of Power Down mode. • 1: The D-Unit constantly drives the CKE pins high to keep the DRAM devices from entering Power Down mode when ranks are idle. Note: This bit is overridden if CKEMODE = 1. This bit does not control CKE behavior on SR entry/exit.

3.97 DRAM Refresh Control (D_CR_DRFC) — Offset 1438h

Specifies the parameters to control scheduling of refresh commands.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1438h	20001750 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	1h RW	**Maintenance Operation Channel Block Time (MNTCHNBLKTIME):** Specifies the amount of time that D-unit will block the channel before any DRAM maintenance operation is issued. (in SPIDclk)
28:25	0h RW	**Maintenance Operation Delay CAS Count (MNTDLYCASCOUNT):** When a critical read request is pending in RPQ and a maintenance operation (MRR, ZQCal, Ref, etc.) needs to be performed, D-unit delays the maintenance operation and allows this many read or write requests to be scheduled before allowing the maintenance operation. Note: This mode does not apply to Panic refreshes.
24:22	0h RO	**Reserved (RSVD24_22):** Reserved
21	0h RW	**Disable Refresh Debt Clear (DISREFDBTCLR):** When set, D-Unit will not clear refresh debt before Self Refresh SR Entry: • 0: D-Unit sends all postponed REF commands to DRAM before it enters Self Refresh. • 1: D-Unit enters SR without clearing the Refresh Debt (for Debug only).

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**Refresh Skew Disable (REFSKWDIS):** Disables Skewing of Refresh Counting between Ranks. Each rank has its own refresh counter. By default incrementing these refresh counters are skewed by 1/2 the tREFI period. Setting this bit to a 1 disables this feature and all refresh counters will increment at the same time per tREFI period. Skewing the tREFI counters can improve performance since traffic to all ranks does not have to be blocked to perform refresh. • 0: Incrementing the refresh counters are skewed by 1/2 tREFI period. • 1: All refresh counters will increment at the same time per tREFI period.
19:18	0h RO	**Reserved (RSVD19_18):** Reserved
17:16	0h RO	**Reserved (RSVD17_16):** Reserved
15	0h RW	**Extra Refresh Debit (EXTRAREFDBT):** When set to 1, D-Unit adds one extra refresh debit (for a total of two) on Self-refresh exit.
14:12	1h RW	**Minimum Refresh Rate (MINREFRATE):** Ensures that refresh rate never drops below a certain limit regardless of TQ polling. • 000: Stop issuing refresh commands and accumulating refresh debits. (does not stop tREFI counter). • 001: 0.25x refresh rate (i.e. 4x tREFI same as no limit). • 010: 0.5x refresh rate (i.e. 2x tREFI). • 011: 1x refresh rate (i.e. 1x tREFI). • 100: 2x refresh rate (i.e. 0.5x tREFI). • 101: 4x refresh rate (i.e. 0.25x tREFI). • 110: 4x refresh rate with derating forced on i.e. 0.25x tREFI. • 111: Reserved.
11:8	7h RW	**Refresh Panic Watermark (REFWMPNC):** When the Refresh counter per rank is greater than this value, the D-Unit will send a REF command to the rank regardless of pending requests. Note: REFWMPNC must be greater than or equal to REFWMHI and greater than 2, Max Value must be less than 8 to not violate 9xtREFI JEDEC requirement.
7:4	5h RW	**Refresh High Watermark (REFWMHI):** When the Refresh counter per rank is greater than this value, the D-Unit will send a REF command to the rank if there is no critical priority requests in the pending queues. Note: Value must be greater or equal to 1 and less than or equal to REFWMPNC.
3:1	0h RO	**Reserved (RSVD3_1):** Reserved
0	0h RW	**Opportunistic Refresh Disable (OREFDIS):** Disable opportunistic scheduling of refresh. • 0: D-Unit will send a REF command only if there is no pending request to that rank. • 1: D-Unit will not send any opportunistic refreshes. Refresh commands are only sent when the refresh counter is greater than REFWMHI. Note: When set, DISREFDBTCLR must also be set to be able to enter SR.

3.98 D-Unit Scheduler Control (D_CR_DSCH) — Offset 143Ch

Specifies parameters to control scheduling of commands to DRAM.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 143Ch	3901C08 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Early Write DQ Enable (EARLY_DQ_EN):** Enables D-unit to send the write data to the PHY one clock earlier than WL value (WL - 1). Note: Applicable to DDR3L and DDR4 only.
30:29	0h RW	**BGF Early Read Data Valid (BGF_EARLY_RDDATA_VALID):** Specifies the number of clocks the D-Unit sends the read data valid through the BGF earlier as compared to the data. • 00: Always write read valid in same SPID clock as data (STATIC_0). • 01: Always write read valid one SPID clock before data (STATIC_1). • 10: Write read valid up to 2 SPID clocks before data (DYNAMIC). • 11: Reserved
28:27	0h RW	**SPID Early Read Data Valid (SPID_EARLY_RDDATA_VALID):** Specifies the delay in SPID clocks from RDDATA_VALID assertion to actual data on SPID. The value should match what is programmed in DDRIO (PHY).
26:21	1Ch RW	**Write Pending Queue Count (WPQCOUNT):** Used to limit the number of available slots in Write Pending Queue/ Write Data Buffer. WPQCOUNT will only recognize changes when PMI ISM is not active.
20:16	10h RW	**Read Pending Queue Count (RPQCOUNT):** Used to limit the number of entries in Read Pending Queue. RPQCOUNT will only recognize changes when PMI ISM is not active.
15	0h RO	**Reserved (RSVD15):** Reserved
14:10	7h RW	**Read Return Data Additional Credits (BLKRDBF_ADD_RDDATA_CR):** Number of additional full cacheline (64B) read data return credits exposed to D-Unit when BLKRDBF is set. Note: The value in this field has no effect on Read return credits when BLKRDBF is not set.
9:8	0h RW	**In-Order Mode (INORDERMODE):** • 0h: In order mode disabled: Commands are sent out of order. • 1h: Partial in order mode: Read and Write CAS commands are sent in the order they were received. ACT and PRE can go out of order. • 2h: Full in order mode serialized test: All DRAM commands CAS ACT PRE associated with a PMI request are issued to DDR before any DRAM commands for a subsequent PMI request. • 3h: Reserved. In order modes should be enabled during init/training/CPGC testing. Should never be changed while the D-Unit queues are nonempty. For WIO2, when in-order mode is enabled (01 or 10), D_CR_DSCH_BYPASSEN must be set to 0
7	0h RW	**Idle Bypass Mode Enable (BYPASSEN):** When set new page hit/empty read requests will bypass D-Unit pipeline stages to save latency 0 - Disable Idle Bypass 1 - Enable Idle Bypass Note: Only applies to WIO2, this bit is reserved for other technologies
6	0h RW	**Block When RDB Full (BLKRDBF):** When set D-Unit stops scheduling new read commands to DRAM when the read data buffer (RDB) is full.
5:4	0h RW	**Stretch Mode (STRETCHMODE):** When stretch mode is enabled, commands are initiated only on Phase 0 of SPIDClk. • 00: Stretch mode is disabled. • 01: Commands are initiated on Phase 0 of every SPID clocks. • 10: Commands are initiated on Phase 0 of even SPID clocks. • 11: Commands are initiated on Phase 0 of odd SPID clocks.
3:0	8h RW	**Masked Write Turnaround Delta (TMWR_TA_DELTA):** The value in this register is subtracted from Masked Write to Read, Masked Write to Write and Masked Write to Masked Write turnaround times to account for half BL MWr commands in LPDDR4 and WIO2. • LPDDR4: = MWr tCCD = MWr BL/2 = 8. • WIO2: = NA.

3.99 DRAM Calibration Control (D_CR_DCAL) — Offset 1440h

Specifies parameters to control ZQ Calibration.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1440h	1057 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**ZQ Calibration Type (ZQCALTYPE):** Determines whether the ZQ Calibration is a long or short calibration command (due to ZQCALSTRT). 0: Short calibration (ZQCS). 1: Long calibration (ZQCL).
30	0h RW/V	**ZQ Calibration Start Rank 1 (ZQCALSTRTR1):** Set this bit to 1 to start the ZQ calibration sequence on Rank 1. This bit will remain a 1 until the ZQ calibration is complete for rank 1, then it will return to 0. 0: ZQ calibration is done. 1: ZQ calibration has started and is in progress.
29	0h RW/V	**ZQ Calibration Start Rank 0 (ZQCALSTRTR0):** Set this bit to 1 to start the ZQ calibration sequence on Rank 0. This bit will remain a 1 until the ZQ calibration is complete for rank 0, then it will return to 0. 0: ZQ calibration is done. 1: ZQ calibration has started and is in progress.
28:23	0h RO	**Reserved (RSVD28_23):** Reserved
22:21	0h RW	**Self-Refresh Exit ZQ Calibration Control (SRXZQC):** • 00: On DynSR exit ZQ timer determines the ZQ type. When the state is lost (i.e due to AutoPG/S0ix) ZQCL is always performed. • 01: Always perform ZQCL after self refresh exit. In LPDDR4, ZQ with traffic blocked. • 10: Always perform ZQCS on SR exit. For LPDDR4, ZQ while traffic is allowed. • 11: No ZQCL commands are sent (it disables ZQCAL commands on SR exit).
20:18	0h RO	**Reserved (RSVD20_18):** Reserved
17	0h RW	**ZQ Calibration Mode (ZQCLMODE):** Specifies how ZQCal commands are sent to different ranks. • 0: ZQCal commands are sent in parallel to all ranks. • 1: ZQCal commands are sent serially to each rank.
16	0h RW	**Periodic ZQ Calibration Disable (ZQCDIS):** • 0: Periodic ZQ Calibration is Enabled. • 1: Disable periodic ZQ Calibration.
15:14	0h RO	**Reserved (RSVD15_14):** Reserved
13:0	1057h RW	**ZQ Calibration Interval (ZQINT):** Specifies the time interval between two ZQCS (LPDDR3/DDR3L/DDR4) or ZQ Start (LPDDR4) commands to a DRAM device. (in RTC 32.8KHz clocks)

3.100 DRAM Mode Registers Shadow Copy (D_CR_MR_SHADOW) — Offset 1448h

This register contains a copy of Mode Regsiters in the DRAM so that D-unit can only modify certain bits without affecting other values.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1448h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:16	0h RW	**MR Value 2 (MR_VALUE2):** MR3 Shadow Register (DDR4): BIOS writes the correct value of MR3 register in DDR4 into this field at boot time. D-Unit modifies one bit and rewrites the MR3 into DDR4 to enter and exit MPR mode.
15:14	0h RO	**Reserved (RSVD15_14):** Reserved
13:0	0h RW	**MR Value (MR_VALUE):** MR3 Shadow Register (WIO2): BIOS sets the value of this field at boot time based on the DRAM device configuration. D-Unit merges the value in MR3_THERM_OFFSET with this field and writes the result into DRAM MR3 MR2 Shadow Register (DDR3L): BIOS writes the correct value of MR2 register in DDR3L into this field at boot time. D-Unit modifies one bit and rewrites the MR2 into DDR3L DRAM before SR entry. MR4 Shadow Register (DDR4): BIOS writes the correct value of MR4 register in DDR4 into this field at boot time. D-Unit modifies one bit and rewrites the MR4 into DDR4 DRAM after temperature read out.

3.101 VNN Scaling Timer Control (D_CR_VNNTIMER) — Offset 144Ch

Specifies parameters for VNN Scaling Timer in D-Unit. The values in this register will be set by P-code during VNN scaling period.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 144Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**VNN Scaling Timer Enable (VNN_TIMER_EN):** • 0: The D-Unit VNN Scaling Timer is disabled. • 1: The D-Unit VNN Scaling Timer is enabled.
30:12	0h RO	**Reserved (RSVD30_12):** Reserved
11:0	0h RW	**VNN Timer Time (VNN_TIMER_TIME):** The final timer value (in 16 x DRAM clocks).

3.102 Periodic DRAM Temperature Polling Control (TQ) (D_CR_TQCTL) — Offset 1450h

Specifies the control for periodic temperature monitoring and control of DRAM device.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1450h	6C000008 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	3h RW/V	**TQ Data Rank 1 (TQDATAR1):** If Rank 1 is disabled, this value will remain zero. This field contains the data of the last temperature sensor read from Rank 1 DRAM Mode Register. It is overwritten with each command.
28:26	3h RW/V	**TQ Data Rank 0 (TQDATAR0):** This field contains the data of the last temperature sensor read from Rank 0 DRAM Mode Register. It is overwritten with each command.
25:22	0h RO	**Reserved (RSVD25_22):** Reserved
21:8	0h RW	**TQ Poll Period (TQPOLLPER):** This sets the frequency by which the D-Unit initiates temperature sensor read from DRAM mode register to determine required refresh rate (in 4x tREFI units).
7	0h RW	**Temperature Controlled Refresh Range Enable (DDR4 Only) (TCRREN):** When set, after a DRAM temperature read out (MPR sequence), D-unit writes a 1 to bit 2 of MR_SHADOW.MR_VALUE when temperature sensor read out for that rank indicates a value higher than 0x1, and writes a 0 to that bit otherwise. The new MR_VALUE is then written into MR4 of DDR4 for each enabled rank.
6	0h RW/V	**TQ Poll Start Rank 1 (TQPOLL_START_R1):** When set Dunit will initiate temperature sensor read for Rank 1. Hardware will clear the bit once the MR read command is issued for this rank. Note: TQPOLLEN mus be 0 before this bit is set.
5	0h RW/V	**TQ Poll Start Rank 0 (TQPOLL_START_R0):** When set Dunit will initiate temperature sensor read for Rank 0. Hardware will clear the bit once the MR read command is issued for this rank. Note: TQPOLLEN mus be 0 before this bit is set.
4	0h RW	**Self Refresh Temperature Range Enable (DDR3 Only) (SRTEN):** When set, before every Self refresh entry, D-Unit writes a 1 to bit 7 of MR_SHADOW.MR_VALUE when TQDATA for that rank indicates a value higher then 0x3, and writes a 0 to that bit otherwise. The new MR_VALUE is then written into MR2 of DDR3 for each enabled rank.
3	1h RW	**Enable Dynamic Timing Derating (ENDERATE):** When set to 1, the Dynamic Timing Derating is enabled. When the D-Unit determines (via TQ polling) that the DRAM requires timing derating in addition to refresh interval adjustment, the D-Unit will automatically adjust the relevant timing parameters.
2	0h RW	**Enable TQ Data Push (TQDATAPUSHEN):** When set to 1, D-Unit pushes the data form the last temperature sensor read to a punit register.
1	0h RW	**Enable TQ Poll on Self-Refresh Exit (TQPOLLSREN):** This bit enables temperature sensor read on Self Refresh Exit. If disabled, D-Unit will not initiate MR Read sequence to read temperature value on Self-Refresh exit.
0	0h RW	**Enable Periodic TQ Poll (TQPOLLEN):** This bit enables periodic TQ Poll. If disabled, D-Unit will not read the DRAM's temperature sensor value periodically.

3.103 Temperature Offset Control (D_CR_TQOFFSET) — Offset 1454h

Specifies temperature offset and refresh rate adjustments requested by software.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1454h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD31_16):** Reserved
15:11	0h RO	**Reserved (RSVD15_11):** Reserved
10:8	0h RW	**MR4 Adder (MR4_ADDER):** D-Unit adds the value of this field to TQDATA read from MR4 the resulting value is used to control refresh rate and AC timing derating.
7:3	0h RO	**Reserved (RSVD7_3):** Reserved
2	0h RW/V	**MR3 Offset Update (MR3_OFFSET_UPDATE):** When set, D-Unit writes the merged value of MR3_VALUE and MR3_THERM_OFFSET into MR3 of DRAM. D-Unit clears this bit once the value is written.
1:0	0h RW	**MR3 Thermal Offset (MR3_THERM_OFFSET):** (WIO2 only) Everytime MR3_OFFSET_UPDATE is set, D-Unit merges the value in this field with MR3_VALUE and writes the result to MR3 in DRAM

3.104 D-Unit Control Operations (D_CR_DCO) — Offset 1458h

Specifies D-Unit initialization and control operation.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1458h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/V	**Initialization Complete (IC):** Indicates that initialization of the D-Unit has been completed. Memory accesses are permitted and maintenance operation begins. Until this bit is set to a 1, the memory controller will not accept DRAM requests from the Bunit/GSA/2LM (PMI ISMs will not leave idle). Note: Set this bit to 1 only when all other D-Unit registers have been configured. Usually set at the last configuration step by BIOS on cold/warm reset. D-Unit hardware sets this bit on SR exit.
30	0h RO/V	**DDRIO PHY Initialization Complete (DIOIC):** Status indication that the DDRIO PHY initialization is complete reflects the status spid_init_complete signal.
29	0h RO	**Reserved (RSVD29):** Reserved
28	0h RW	**PMI Control Select (PMICTL):** • 0: D-Unit PMI is connected to Bunit/GSA/2LM. • 1: D-Unit PMI is connected to CPGC. Note: D_CR_DSCH_BYPASSEN must be set to 0 in CPGC mode. Note: PMI must be idle and D-Unit BGF_RUN = 0 before changing the value in this register.
27:8	0h RO	**Reserved (RSVD27_8):** Reserved
7:4	0h RO	**Reserved (RSVD7_4):** Reserved
3	0h RW	**Enable PSMI Mode (PSMIEN):** When enabled, D-Unit will synchronize clock crossing signals. • 0: PSMI Mode is disabled. • 1: PSMI Mode is enabled. Note: Change only allowed when D-Unit is idle.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Maintenance Reset (MNTRST):** Writing a 1 to this field resets all maintenance timers. Clears all states and also clears refresh debt queues. This bit must be cleared by software after at least 3 SPID clocks.
1	0h RW	**Enable Maintenance Operations (MNTEN):** Setting this field to 1 enables all maintenance operations. When DCO.IC is set, the maintenance operations are enabled irrespective of the value of this field.
0	0h RO	**Reserved (RSVD0):** Reserved

3.105 Data Scrambler (D_CR_SCRAMCTRL) — Offset 14A4h

Specifies parameters to control data scrambling in D-Unit.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 14A4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Enable Data Scrambler (SCRM_EN):** When set to 1, data scrambling is enabled. When set to 0, data scrambling is disabled. Should be set before D_CR_BGF_CTL_BGF_RUN is set to 1.
30	0h RO	**Reserved (RSVD30):** Reserved
29:28	0h RW	**Scrambler Clock Gate Select (CLOCKGATE):** This field controls how the scrambler output code is clock gated to reduce power. • 00: Clock gate disabled. • 01: Clock Gate every 2 cycles. • 10: Clock Gate every 3 cycles. • 11: Clock Gate every 4 cycles.
27:16	0h RO	**Reserved (RSVD27_16):** Reserved
15:0	0h RW	**Scrambling Key (KEY):** Sets the key for the scrambler. The key should be a random value that is set following each cold boot.

3.106 Error Injection Address Register (D_CR_ERR_INJ) — Offset 14ACh

Contains the target address for ECC error injection.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 14ACh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD31):** Reserved
30:1	0h RW	**Error Injection Target Address (ADDRESS):** Specifies the PMI address of the write transaction to be injected with the error. Only applicable to Write transactions. Read/ under-fill read of the partial write operation is not affected.
0	0h RO	**Reserved (RSVD0):** Reserved

3.107 Error Injection Control Register (D_CR_ERR_INJ_CTL) — Offset 14B0h

Controls injecting correctable or uncorrectable errors into the write requests specified by target address.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 14B0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RSVD31_4):** Reserved
3	0h RW	**Error Injection Type Higher 32B (SEL_HI):** 0 - Uncorrectable Error (UE) is armed for write address matching to inject UE by using the same poisoning scheme, i.e. inverting corresponding write ECC[6:0] on every QW of the 32B data. 1 - Correctable Error (CE) is armed for write address matching to inject CE by inverting corresponding write ECC[0] on every QW of the 32B data.
2	0h RW	**Error Injection Enable Higher 32B (EN_HI):** When set the error injection is continuously armed for higher 32B of D_CR_ERR_INJ_ADDR write address matching until it is cleared.
1	0h RW	**Error Injection Type Lower 32B (SEL_LO):** 0 - Uncorrectable Error (UE) is armed for write address matching to inject UE by using the same poisoning scheme, i.e. inverting corresponding write ECC[6:0] on every QW of the 32B data. 1 - Correctable Error (CE) is armed for write address matching to inject CE by inverting corresponding write ECC[0] on every QW of the 32B data.
0	0h RW	**Error Injection Enable Lower 32B (EN_LO):** When set, the error injection is continuously armed for lower 32B of D_CR_ERR_INJ_ADDR write address matching until it is cleared.

3.108 Error Log Register (D_CR_ERR_ECC_LOG) — Offset 14B4h

Detected ECC errors are captured in this register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 14B4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/V	**CLEAR:** Setting this bit to one clears all fields in this register, including itself.
30:29	0h RW	**PMI VISA Byte Select (ECC_VISA):** Select ECC or PMI byte on VISA : • 00: ECC byte, • 01: PMI Data Byte [7:0], • 10: PMI Data Byte [63:56], • 11: PMI Data Byte [255:248]
28	0h RW/V	**Correctable Single-bit Error (CERR):** This bit is set when a correctable single-bit error occurs on a memory read data transfer. When this bit is set, the address that caused the error and the error syndrome are also logged and they are locked to further single bit errors, until this bit is cleared. A multiple bit error that occurs after this bit is set will override the address/error syndrome information.
27	0h RW/V	**Uncorrectable Multiple-bit Error (MERR):** This bit is set when an uncorrectable multiple-bit error occurs on a memory read data transfer. When this bit is set, the address that caused the error and the error syndrome are also logged and they are locked until this bit is cleared.
26:25	0h RW/V	**Error Burst Number (ERR_BURST):** Burst number (in BL8) of the error within a chunk.
24	0h RW/V	**Error Chunk Number (ERR_CHUNK):** Chunk number of the error. 0 - lower 32B chunk has error if MERR/CERR is set 1 - higher 32B chunk has the error if MERR/CERR is set
23:16	0h RW/V	**Quad Word ECC Syndrome (SYNDROME_QW):** ECC Syndrome for a QW (64 bit) within 32B Address
15:0	0h RW/V	**Request Tag (TAG):** Read Return Tag matches with the PMI Request Tag which triggered the error log.

3.109 D-Unit Fuse Status (D_CR_DFUSESTAT) — Offset 14BCh

Contains the values read from D-Unit fuses.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 14BCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD31_16):** Reserved
15:0	0h RO/V	**D-Unit Fuse Status (FUSESTAT):** D-Unit fuse bits are captured into this register and are available to be read. • [0]: fus_dun_ecc_dis. • [3:1]: fus_dun_max_supported_device_size[2:0]. • [4:4]: fus_dun_lpddr3_dis. • [5:5]: fus_dun_lpddr4_dis. • [6:6]: fus_dun_wio2_dis. • [7:7]: fus_dun_ddr3l_dis. • [8:8]: fus_dun_ddr4_dis. • [15:9]: reserved.

3.110 Major Mode Control (D_CR_MMC) — Offset 1524h

Specifies parameters to control read/write major mode operation and transitions.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1524h	2B01A518 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:27	5h RW	**RAW Conflict Read Priority for WMM Transition (RAW_WMM):** If a conflict read reaches this priority (or greater depending on access class occupancy), WMM will be triggered to unblock the corresponding write. D-Unit will stay in WMM until corresponding write is issued. Note: The value in this bit must not be higher than lowest terminal priority level of each access class.
26	0h RO	**Reserved (RSVD26):** Reserved
25:23	6h RW	**Read Isoch Trigger Priority (RIMPRIO):** If any read in the RPQ is at this programmable priority, RIM is triggered.
22:18	0h RO	**Reserved (RSVD22_18):** Reserved
17:12	1Ah RW	**Write Isoch Threshold (WIMTHRS):** When the number of entries in WPQ is greater than or equal to this value (higher than WMM entry watermark, less than WPQ size), it triggers write isoch mode (WIM).
11:6	14h RW	**Write Major Mode Exit Watermark (WMMEXIT):** When the number of entries in WPQ is less than this value, the D-Unit will switch back to read major mode.
5:0	18h RW	**Write Major Mode Entry Watermark (WMMENTRY):** When the number of entries in WPQ is greater than or equal to this value, the D-Unit will switch to write major mode (WMM). Note: the value must not be set to 0.

3.111 Major Mode RD/WR Counter (Set A and B) (D_CR_MMRDWR_AB) — Offset 1528h

Minimum read and maximum write counter control. This register defines the minimum number of reads in RMM and maximum number of writes in WMM before a mode transition happens.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1528h	1020408 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (RSVD31_26):** Reserved
25:20	10h RW	**Max Writes B (MAXWRB):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set B).

Bit Range	Default & Access	Field Name (ID): Description
19:14	8h RW	**Min Reads B (MINRDB):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set B).
13:12	0h RO	**Reserved (RSVD13_12):** Reserved
11:6	10h RW	**Max Writes A (MAXWRA):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set A).
5:0	8h RW	**Min Reads A (MINRDA):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set A).

3.112 Major Mode RD/WR Counter (Set C and D) (D_CR_MMRDWR_CD) — Offset 152Ch

Minimum read and maximum write counter control. This register defines the minumum number of reads in RMM and maximum number of writes in WMM before a mode transition happens (sets C and D).

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 152Ch	1020408 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (RSVD31_26):** Reserved
25:20	10h RW	**Max Writes D (MAXWRD):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set D).
19:14	8h RW	**Min Reads D (MINRDD):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set D).
13:12	0h RO	**Reserved (RSVD13_12):** Reserved
11:6	10h RW	**Max Writes C (MAXWRC):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set C).
5:0	8h RW	**Min Reads C (MINRDC):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set C).

3.113 Access Class Initial Priority (D_CR_ACCIP) — Offset 1530h

Each field of this register defines the initial priority of one access class.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1530h	17C2 h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD31_15):** Reserved
14:12	1h RW	**Access Class 4 Initial Priority (AC4IP):** Initial priority level of read requests coming with access class 4.
11:9	3h RW	**Access Class 3 Initial Priority (AC3IP):** Initial priority level of read requests coming with access class 3.
8:6	7h RW	**Access Class 2 Initial Priority (AC2IP):** Initial priority level of read requests coming with access class 2.
5:3	0h RW	**Access Class 1 Initial Priority (AC1IP):** Initial priority level of read requests coming with access class 1.
2:0	2h RW	**Access Class 0 Initial Priority (AC0IP):** Initial priority level of read requests coming with access class 0.

3.114 Access Class 0 Priority Promotion Control (D_CR_RD_PROM0) — Offset 1534h

This register defines the priority promotion policy for access class 0. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1534h	1F52940 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
24:20	1Fh RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	Ah RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	Ah RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	Ah RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.115 Access Class 1 Priority Promotion Control (D_CR_RD_PROM1) — Offset 1538h

This register defines the priority promotion policy for access class 1. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the associated level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1538h	14000000 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	Ah RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
24:20	0h RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	0h RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	0h RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	0h RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.116 Access Class 2 Priority Promotion Control (D_CR_RD_PROM2) — Offset 153Ch

This register defines the priority promotion policy for access class 2. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 153Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

Bit Range	Default & Access	Field Name (ID): Description
24:20	0h RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	0h RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	0h RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	0h RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.117 Access Class 3 Priority Promotion Control (D_CR_RD_PROM3) — Offset 1540h

This register defines the aging policy for access class 3. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1540h	1F29400 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
24:20	1Fh RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	5h RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	5h RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	0h RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.118 Access Class 4 Priority Promotion Control (D_CR_RD_PROM4) — Offset 1544h

This register defines the aging policy for access class 3. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1544h	1F5294A h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
24:20	1Fh RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	Ah RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	Ah RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	Ah RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	Ah RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.119 Deadline Threshold (D_CR_DL_THRS) — Offset 1548h

Specifies when the request with initial priority 0 get promoted to a higher priority level.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1548h	6 h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD31_11):** Reserved
10:0	6h RW	**Deadline Threshold (DEADLINE_THRS):** A requests with initial priority of 0 will exit priority 0 when its deadline is equal or less than this value plus current time. This field does not affect the priority of any requests in access classes with initial priority bigger than 0.

3.120 Major Mode Blocking Rules Control (D_CR_MM_BLK) — Offset 154Ch

This register controls blocking rules enforced in RMM and WMM.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 154Ch	1800 h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD31_25):** Reserved
24	0h RW	**WMM Regular Rule 1 (WMM_REG_R1):** Disable WMM unsafe write page hits block safe write page misses same bank.
23:20	0h RO	**Reserved (RSVD23_20):** Reserved
19	0h RW	**WMM Priority Rule 4 (WMM_PRIO_R4):** Disable WMM unsafe priority 1 read miss block write hit to same bank. Note: This rule does not block the bank that is being blocked by WMM_PRIO_R3. Priority rules 1, 3 and 4 should be enabled/disabled together.
18	0h RW	**WMM Priority Rule 3 (WMM_PRIO_R3):** Disable WMM unsafe priority 1 write hit block write miss to same bank. Note: This rule does not block the bank that is being blocked by WMM_PRIO_R1. Priority rules 1,3 and 4 should be enabled/disabled together.
17	0h RW	**WMM Priority Rule 2 (WMM_PRIO_R2):** Disable WMM CAS block rule.
16	0h RW	**WMM Priority Rule 1 (WMM_PRIO_R1):** Disable WMM unsafe top priority 1 write miss block write hit same bank. Priority rules 1, 3 and 4 should be enabled/disabled together.
15:14	0h RO	**Reserved (RSVD15_14):** Reserved
13	0h RW	**RMM Regular Rule 6 (RMM_REG_R6):** Disable RMM unsafe write page hits block safe write page misses same bank.
12	1h RW	**RMM Regular Rule 5 (RMM_REG_R5):** Disable RMM unsafe read page miss block all safe and unsafe write page hit to the same bank. Note: This field must not be set to 0 (enabled) if RMM_REG_R4 is also 0.
11	1h RW	**RMM Regular Rule 4 (RMM_REG_R4):** Disable RMM unsafe write page hit block safe read page miss same bank. Note: This field must not be set to 0 (enabled) if RMM_REG_R5 is also 0.
10	0h RW	**RMM Regular Rule 3 (RMM_REG_R3):** Disable RMM unsafe read page hit block safe read and write page miss same bank. Note: This rule does not block the bank that is being blocked by RMM_PRIO_R3 and RMM_PRIO_R1.
9	0h RW	**RMM Regular Rule 2 (RMM_REG_R2):** Disable RMM unsafe read page empty block safe write page empty same rank.
8	0h RW	**RMM Regular Rule 1 (RMM_REG_R1):** Disable RMM unsafe read page hit block safe write page hit same rank.
7:4	0h RO	**Reserved (RSVD7_4):** Reserved
3	0h RW	**RMM Priority Rule 4 (RMM_PRIO_R4):** Disable RMM unsafe critical read miss block read and write hit to same bank. Note: This rule does not block the bank that is being blocked by RMM_PRIO_R3. Priority rules 1, 3 and 4 should be enabled/disabled together.
2	0h RW	**RMM Priority Rule 3 (RMM_PRIO_R3):** Disable RMM unsafe critical read hit block read and write miss to same bank. Note: This rule does not block the bank that is being blocked by RMM_PRIO_R1. Priority rules 1, 3 and 4 should be enabled/disabled together.
1	0h RW	**RMM Priority Rule 2 (RMM_PRIO_R2):** Disable RMM CAS block rule.
0	0h RW	**RMM Priority Rule 1 (RMM_PRIO_R1):** Disable RMM unsafe top critical read miss block read and write hit same bank. Note: Priority rules 1, 3 and 4 should be enabled/disabled together.

3.121 DRAM Self-Refresh Command (D_CR_DRAM_SR_CMD) — Offset 1554h

Self refresh command register to allow sending WAKE and SUSPEND messages to D-Unit. (Only one bit can be set at a time). Posted writes to this register are not completed until hardware clears the field.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1554h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RSVD31_4):** Reserved
3	0h RW/V	**SUSPENDP:** A SUSPENDP message will put the DRAM into self-refresh mode. The D-Unit will complete servicing outstanding memory requests and flush all queued Refresh commands to DRAM before putting the DRAM in self refresh mode. Finally, a PM message will be sent to the PHY. The bit is cleared by hardware after the PHY indicates the transition requested in the PM message has been completed. D-Unit will perform an MRW to MR17 with an opcode as defined by DPMC0.PASR before it places the DRAM into Self-Refresh.
2	0h RW/V	**SUSPEND:** A SUSPEND message will put the DRAM into self-refresh mode. The D-Unit will complete servicing outstanding memory requests and flush all queued Refresh commands to DRAM before putting the DRAM in Self Refresh mode. Finally, a PM message will be sent to the PHY. The bit is cleared by hardware only after the PHY indicates the transition requested in the PM message has been completed. Note: When COLDWAKE is set prior of setting this bit the DRAM will not be placed in SR.
1	0h RO	**Reserved (RSVD1):** Reserved
0	0h RW/V	**WAKE:** Take PHY out of PM states and wakes the DRAM out of self refresh mode. The bit is cleared by hardware only when the DRAM has exited out of self refresh mode and is accessible. Note: When COLDWAKE is set prior of setting this bit the D-Unit will not send SR exit command and will not set the DCO.IC bit.

3.122 DQS Retraining Control (D_CR_DQS_RETRAINING_CTL) — Offset 1580h

LPDDR4 DQS Retraining control register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1580h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DQS Periodic Retraining Interval (DQS_RETRAIN_INT):** This sets the frequency by which the D-Unit initiates periodic retraining (in 1x NREFI).
15:14	0h RO	**Reserved (RSVD15_14):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:4	0h RW	**DQS Oscillator Runtime (DQS_OSC_RT):** After D-Unit starts DQS oscillator, it must wait this amount of time before being able to read the value in MR18 and MR19 (in 16x DRAM clocks). Value in this register must be at least equal to DRAM's MR23 value. + tOSCO.
3	0h RW/V	**DQS Retrain Start Rank 1 (DQS_RETRAIN_START_R1):** When set, Dunit will initiate LPDDR4 DQS training sequence on Rank1. Hardware will clear bit when sequence completes (after issuing MR19). Note: DQS_RETRAIN_EN must be 0 before setting this bit.
2	0h RW/V	**DQS Retrain Start Rank 0 (DQS_RETRAIN_START_R0):** When set, Dunit will initiate LPDDR4 DQS training sequence on Rank0. Hardware will clear bit when sequence completes (after issuing MR19). Note: DQS_RETRAIN_EN must be 0 before setting this bit.
1	0h RW	**DQS Retrain SRX Exit (DQS_RETRAIN_SRX_EN):** Enable retraining on SR exit. This bit enables LPDDR4 DQS retraining on Self Refresh Exit. If disabled, D-Unit will not perform retraining on SR exit.
0	0h RW	**DQS Retrain Enable (DQS_RETRAIN_EN):** Periodic retraining enable: This bit enables periodic DQS retraining. If disabled, D-Unit will not perform retraining periodically. Note: Will be enabled only if DCO.IC is set and refreshes are enabled in DRF.MINREFRATE.

3.123 MR4 De-Swizzle Control (D_CR_MR4_DESWIZZLE) — Offset 1584h

Controls the data bits swizzling crossbar for MR4.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1584h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD31):** Reserved
30:28	0h RW	**MR4 Bit 2 Select 2nd Byte (MR4_BIT2_SEL2):** Selects bit 2 of MR4 data.
27	0h RO	**Reserved (RSVD27):** Reserved
26:24	0h RW	**MR4 Bit 1 Select 2nd Byte (MR4_BIT1_SEL2):** Selects bit 1 of MR4 data
23	0h RO	**Reserved (RSVD23):** Reserved
22:20	0h RW	**MR4 Bit 0 Select 2nd Byte (MR4_BIT0_SEL2):** Selects bit 0 of MR4 data.
19:18	0h RO	**Reserved (RSVD19_18):** Reserved
17:16	0h RW	**MR4 Byte 2 Select (MR4_BYTE_SEL2):** Selects byte position of the MR4 data for second device.
15	0h RO	**Reserved (RSVD15):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
14:12	0h RW	**MR4 Bit 2 Select (MR4_BIT2_SEL):** Selects bit 2 of MR4 data.
11	0h RO	**Reserved (RSVD11):** Reserved
10:8	0h RW	**MR4 Bit 1 Select (MR4_BIT1_SEL):** Selects bit 1 of MR4 data.
7	0h RO	**Reserved (RSVD7):** Reserved
6:4	0h RW	**MR4 Bit 0 Select (MR4_BIT0_SEL):** Selects bit 0 of MR4 data.
3:2	0h RO	**Reserved (RSVD3_2):** Reserved
1:0	0h RW	**MR4 Byte Select (MR4_BYTE_SEL):** Selects byte position of the MR4 data first device.

3.124 DRAM Rank Population 0 (D_CR_DRP0) — Offset 1600h

Rank configuration register.

This register is start of dunit 11 LPDDR4 Channel 3, DDR4 Channel 1.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1600h	10000000 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RW	**DRAM Device Per Rank (DRAMDEVICE_PR):** Specifies the number of DRAM devices that are ganged together to form a single rank. • 00: 1 DRAM device in each rank. • 01: 2 DRAM devices in each rank. • 10: 4 DRAM devices in each rank. • 11: 8 DRAM devices in each rank. Note: The actual number of devices is one more than the value programmed when ECC is enabled.
29:28	1h RW	**Address Decode (ADDRDEC):** Specifies the address mapping to be used: • 00: 1KB (A). • 01: 2KB (B). • 10: 4KB (C). • 11: Reserved.
27:25	0h RW	**Burst Length Mode (BLMODE):** • 000: Fixed BL8. • 001: Onthefly BL8. • 010: Fixed BL16. • 011: Onthefly BL16. • 100: Fixed BL32. • 101: Onthefly BL32. • 110-111: Reserved.

Bit Range	Default & Access	Field Name (ID): Description
24:22	0h RW	**DRAM Type (DRAMTYPE):** • 000: Reserved. • 001: Reserved. • 010: LPDDR4. • 011: Reserved. • 100: DDR4. • 101-111: Reserved. Note: The D-Unit should only use this field if allowed by fuse.
21	0h RW	**ECC Enable (ECCEN):** • 0: ECC is disabled. • 1: ECC is enabled. This bit determines if the D-Unit treats the PMI BE_ECC bits as ECC bits or Byte Enables. The D-Unit should not allow this bit to be set if ECC is disabled by fuse. This should only be used in configurations that support ECC.
20:19	0h RW	**CA Swizzle Type (CASWIZZLE):** • 00: uniDIMM/SODIMM/UDIMM. • 01: BGA. • 10: NA. • 11: uniDIMM/SODIMM/UDIMM with Rank 1 mirrored (DDR4 Only).
18:17	0h RO	**Reserved (RSVD18_17):** Reserved
16	0h RW	**Fine Bank Group Interleaving Enable (FBGINTEN):** When enabled, D-unit will use a lower order address bit in the bank hashing function (DDR4 Only). • 0: Bank Group Interleave disabled. • 1: Bank Group Interleave enabled. Note: BAHEN must be set for this bit to take effect.
15	0h RW	**Bank Address Hashing Enable (BAHEN):** • 0: Bank Address Hashing disabled. • 1: Bank Address Hashing enabled.
14	0h RW	**Rank Select Interleave Enable (RSIEN):** • 0: Rank Select Interleaving disabled. • 1: Rank Select Interleaving enabled.
13:9	0h RO	**Reserved (RSVD13_9):** Reserved
8:6	0h RW	**DRAM Device Density (DDEN):** Density of the DRAM devices populated on Ranks 0 and 1. • 000: 4 Gb. • 001: 6 Gb. • 010: 8 Gb. • 011: 12 Gb. • 100: 16 Gb. • 101-111: Reserved. Note: For LPDDR4 this value is the die density.
5:4	0h RW	**DRAM Device Data Width (DWID):** Data width of the DRAM device populated on Ranks 0 and 1. • 00: x8. • 01: x16. • 10: x32. • 11: x64.
3	0h RO	**Reserved (RSVD3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Dual Data Mode Enable (DDMEN):** • 0: PMI Dual Data Mode is disabled in D-Unit. • 1: PMI Dual Data Mode is enabled. Note: Dual Data Mode must be enable for DDR3L/DDR4 configurations.
1	0h RW	**Rank Enable 1 (RKEN1):** Enable Rank 1: Must be set to 1 to enable use of this rank.
0	0h RW	**Rank Enable 0 (RKEN0):** Enable Rank 0: Must be set to 1 to enable use of this rank. Note: Setting this bit to 0 is not a functional mode.

3.125 DRAM Timing Register 9A (D_CR_DTR9A) — Offset 1604h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1604h	14A546 h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RO	**Reserved (RSVD31_23):** Reserved
22:18	5h RW	**Write to Write DQ Delay Same Bank Group Same Rank (TWWSR_L):** Specifies the delay from a DRAM Write to another Write command within the same bank group of the same rank (in DRAM clocks). DDR4 Equation: tWWSR_L = tCCD_L.
17:13	5h RW	**Read to Read DQ Delay Same Bank Group Same Rank (TRRSR_L):** Specifies the delay from a DRAM Read to another Read command within the same bank group of the same rank (in DRAM clocks). DDR4 Equation: tRRSR_L = tCCD_L.
12:6	15h RW	**Write to Read DQ Delay Same Bank Group Same Rank (TWRSR_L):** Specifies the delay from a DRAM Read to Write command within the same bank group of the same rank (in DRAM clocks). • DDR4 Equation: tWRSR_L = CWL + tDQSSmax + BL/2 + tWPST + tWTR_L.
5:0	6h RW	**Row Activation to Row Activation to Same Bank Group Delay [tRRD_L] (TRRD_L):** Specifies the minimum delay (in DRAM clocks) between two DRAM Activate commands to different banks within the same bank group of a rank. (DDR4 Only)

3.126 DRAM Timing Register 0A (D_CR_DTR0A) — Offset 1608h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1608h	820702CB h

Bit Range	Default & Access	Field Name (ID): Description
31:27	10h RW	**Valid Clocks Before CKE High [tCKCKEH/tCSCKEH/tCKSRX] (TCKCKEH):** Number of valid clocks before CKE high (in DRAM clocks). • LPDDR4: The value in this register covers both tCKCKEH and tCSCKEH. • DDR3L/DDR4/LPDDR3/WIO2: The value covers tCKSRX which is defined as the number of valid DRAM clocks that have to toggle before the issuing of the Self Refresh Exit SRX. This value is also used if the clock frequency is changed or the clock is stopped or tristated during Power Down i.e. the number valid DRAM clocks that have to toggle before the issuing of the Power Down Exit PDX command. TCKCKEH can be used to compensate for clock stabilization delays in the motherboard. Note: D-unit hardware enforces minimum of two SPID clock before CKEH, any value in this register is the additional time.
26:22	8h RW	**Exit Self-Refresh to Valid Commands Requiring a Locked DLL Delay [tXSDLL] (TXSDLL):** D-Unit waits max(tXSR+tZQCL/tZQCS, tXSDLL) before allowing traffic to DRAM (in 32 x DRAM Clocks). LPDDR3/LPDDR4/WIO2: tXSDLL = 0. DDR3L/DDR4: tXSDLL = tDLLK. Note: In the equation above, tZQCL/tZQCS = 0 if no ZQ is performed on SR exit.
21:12	70h RW	**Exit Self-Refresh to Valid Command Delay [tXS/tXSR] (TXSR):** DDR3L/DDR4: tXS - Delay between Self Refresh Exit SRX to any DRAM Command not requiring DLL Lock. LPDDR/WIO2: tXSR - Delay between Self Refresh Exit SRX to any DRAM Command. (in DRAM clocks).
11:6	Bh RW	**Activate RAS to CAS Command Delay [tRCD] (TRCD):** Specifies the delay between a DRAM Activate command and a DRAM Read or Write command to the same bank (in DRAM clocks). Note: Derating adds 1.875ns to this timing.
5:0	Bh RW	**Precharge to Activate Command Delay of a Single Bank [tRPpb] (TRPPB):** Specifies the delay between a DRAM Precharge command and a DRAM Activate command to the same bank (in DRAM Clocks). Note : this CR should be constrained to a minimum of 4 in LPDDR3/DDR3L/WIO2 and minimum of 8 in LPDDR4/DDR4. Note: Derating adds 1.875ns to this timing.

3.127 DRAM Timing Register 1A (D_CR_DTR1A) — Offset 160Ch

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 160Ch	30481218 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	6h RW	**Exit Power Down to Next Command Delay [tXP] (TXP):** Specifies the delay from the DRAM Power Down Exit (PDX) command to any valid command (in DRAM clocks). Note: The value in this field must be programmed to tXPDLL when Slow Exit Mode Power-down is enabled for DDR3L.
26	0h RO	**Reserved (RSVD26):** Reserved
25:14	120h RW	**ZQ (long) Calibration Time [tZQCL/tZQCAL] (TZQCL):** • LPDDR3/DDR3L/DDR4: tZQCL/tZQoper - Specifies the delay between the DRAM ZQ Calibration Long (ZQCL) command and any DRAM command during normal operation. • LPDDR4: tZQCAL - ZQ Calibration time (in DRAM clocks). Note: This field defines the ZQ Calibration Long delay during normal operation. It is not the same as tZQinit which uses the same ZQCL command but the delay is longer. tZQinit applies only during poweron initialization of the DRAM devices and tZQoper applies during normal operation. BIOS executes the DRAM initialization sequence so it has to ensure tZQinit is met and not the D-Unit.

Bit Range	Default & Access	Field Name (ID): Description
13:6	48h RW	**ZQ Short Calibration Time [tZQCS] (TZQCS):** ZQCS to any DRAM Command Delay: Specifies the delay between the DRAM ZQ Calibration Short (ZQCS) command and any DRAM command (in DRAM clocks). DDR3L/DDR4/LPDDR3 only. LPDDR4 does not support ZQCS command
5:0	18h RW	**ZQ Latch Time [tZQLAT] (TZQLAT):** Specifies the delay between the DRAM ZQ Calibration Latch command and any DRAM command (in DRAM clocks). LPDDR4 only. Not used in DDR3L/DDR4/LPDDR3/WIO2.

3.128 DRAM Timing Register 2A (D_CR_DTR2A) — Offset 1610h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1610h	46080C30 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	118h RW	**All Bank Refresh Cycle Time [tRFCab] (NRFCAB):** Specifies the delay between the REFab command to the next valid command. (in DRAM clocks)
21	0h RO	**Reserved (RSVD21):** Reserved
20:17	4h RW	**CKE Minimum Pulse Width [tCKE] (TCKE):** Specifies the minimum time from CKEL to CKEH (in DRAM clocks).
16	0h RO	**Reserved (RSVD16):** Reserved
15:0	C30h RW	**Refresh Interval Time [tREFI] (NREFI):** Specifies the average time between refresh commands. JEDEC Base Refresh Interval time (in DRAM clocks). Note: D-Unit will ignore the 2 LSBs of this field.

3.129 DRAM Timing Register 3A (D_CR_DTR3A) — Offset 1614h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1614h	3002EA28 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	6h RW	**Read to Precharge Delay [tRDPRE] (TRTP):** Specifies the minimum delay between the DRAM Read and Precharge commands to the same bank (in DRAM clocks). • LPDDR3 Equation: = BL/2 + tRTP - 4. • LPDDR4 Equation: = BL/2 + Max (8, tRTP) - 8. • WIO2 Equation: = NA. • DDR3L/DDR4 Equation: = tRTP.

Bit Range	Default & Access	Field Name (ID): Description
26:20	0h RW	**CAS to CAS Command Delay Adder (TCCD_INC):** Specifies the number of clocks to be added to turnaround times (for Stretch Mode). It increases delay between Read to Read or Read to Write commands by the amount programmed in this field (in 4 x DRAM clocks).
19:13	17h RW	**Write to Precharge Command Delay [tWRPRE] (TWTP):** Specifies the minimum delay between the DRAM Write command and the Precharge command to the same bank (in DRAM clocks). • LPDDR3/LPDDR4/WIO2 Equation: tWTP = BL/2 + WL + tWR + 1. • DDR3L/DDR4 Equation: tWTP = BL/2 + CWL + tWR.
12:11	1h RW	**DRAM Command Valid Duration (TCMD):** Specifies the number of DRAM clocks a command is held valid on the DRAM Address and Control buses. 1N is the DDR3 basic requirement. 2N is the extended mode for board signal integrity. • 0h: Reserved. • 1h: 1 DRAM Clock (1N). • 2h: 2 DRAM Clocks (2N). • 3h: Reserved. Note: DDR3L/DDR4 only. tCMD must be set to 1N for LPDDR3/LPDDR4/WIO2.
10:6	8h RW	**Write Latency [WL/CWL] (TCWL):** The delay between the internal write command and the availability of the first word of DRAM input data (in DRAM clocks).
5:0	28h RW	**Write CAS to Masked Write CAS Delay Same Bank (TWMWSB):** Specifies the minimum delay between DRAM Write command to Masked Write command to same bank (in DRAM clocks). • LPDDR4 Equation: tWMWSB = tCCDMW (BL16) or tCCDMW + 8 (BL32). • WIO2 Equation: NA. Note: Masked Write operation in LPDDR4 is always BL16 and in WIO2 is always BL4. D-Unit applies this timing for same rank as well as same bank.

3.130 DRAM Timing Register 4A (D_CR_DTR4A) — Offset 1618h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1618h	30209149 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	30h RW	**Four Bank Activate Window [tFAW] (TFAW):** A rolling timeframe in which a maximum of four Activate commands can be issued to the same rank. This is to limit the peak current draw from the DRAM devices (in DRAM clocks).
23:18	8h RW	**Write to Read DQ Delay Different Ranks (TWRDR):** Specifies the delay from the start of a Write data burst of one rank to the start of a Read data burst of a different rank (in DRAM clocks). • LPDDR3 Equation: tWRDR = WL + tDQSSmax + BL/2 - (RL + tDQSCKmin - tRPRE). • LPDDR4 Equation: tWRDR = WL - RL + BL/2 + 4 - tDQSCKmin. • DDR3L/DDR4 Equation: tWRDR = CWL + tDQSSmax + BL/2 - (CL + tDQSCKmin - tRPRE). Note: For LPDDR3/4 using ODT, this latency may need to be increased by tODToffadj.

Bit Range	Default & Access	Field Name (ID): Description
17:12	9h RW	**Read to Write DQ Delay Different Ranks (TRWDR):** Specifies the delay from the start of a Read data burst of one rank to the Start of a Write data burst of a different rank (in DRAM clocks). • LPDDR3/LPDDR4 Equation: tRWDR = RL + tDQSCKmax + BL/2 - (WL - tWPRE). • DDR3L/DDR4 Equation: tRWDR = CL + tDQSCKmax + BL/2 - (CWL - tWPRE). Note: For LPDDR3/4 using ODT, this latency may need to be adjusted by tODTon. Note: For DDR3L/DDR4 using ODT, this latency may need to be increased by one clock.
11:6	5h RW	**Write to Write DQ Delay Different Ranks (TWWDR):** Specifies the delay from the start of a Write data burst of one rank to the start of a Write data burst of a different rank (in DRAM clocks). • LPDDR3/DDR3L/DDR4 Equation: tWWDR = BL/2 + tDQSSmax - tDQSSmin + tWPRE. • LPDDR4 Equation: tWWDR = BL/2 + 4 - tDQSSmin. Note: For LPDDR3/4 using ODT, this latency may need to be increased by tODToffadj.
5:0	9h RW	**Read to Read DQ Delay Different Ranks (TRRDR):** Specifies the delay from the start of a Read data burst of one rank to the Start of a Read data burst of a different rank (in DRAM clocks). • Equation: tRRDR = BL/2 + tDQSCKmax - tDQSCKmin + tRPRE.

3.131 DRAM Timing Register 5A (D_CR_DTR5A) — Offset 161Ch

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 161Ch	304200C2 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	6h RW	**Row Activation to Row Activation Delay [tRRD] (TRRD):** Specifies the minimum delay in DRAM clocks between two DRAM Activate commands to the same rank but different banks (tRC is the minimum delay between activations of the same bank). Note: Derating adds 1.875ns to this timing.
26	0h RO	**Reserved (RSVD26):** Reserved
25:23	0h RW	**Derate Increment (TDERATE_INC):** Specifies the additional delay that is added to DRAM timing when indicated by MR4 status. (in DRAM clocks) LPDDR3/LPDDR4/WIO2: Value is 1.875ns. Note: The value in this register is only added to these timing parameters: tRCD, tRAS, tRP and tRRD.
22:18	10h RW	**Write to Write DQ Delay Same Rank (TWWSR):** Specifies the delay from a DRAM Write to another Write command of the same rank (in DRAM clocks). Equation: tWWSR = tCCD.
17:13	10h RW	**Read to Read DQ Delay Same Rank (TRRSR):** Specifies the delay from a DRAM Read to another Read command of the same rank (in DRAM clocks). Equation: tRRSR = tCCD.
12:6	3h RW	**Write to Read DQ Delay Same Rank (TWRSR):** Specifies the delay from a DRAM Read to Write command of the same rank (in DRAM clocks). • LPDDR3/LPDDR4/WIO2 Equation: tWRSR = WL + tDQSSmax + BL/2 + tWTR. • DDR3L Equation: tWRSR = CWL + tDQSSmax + BL/2 + tWTR. • DDR4 Equation: tWRSR = CWL + tDQSSmax + BL/2 + tWTR_S.

Bit Range	Default & Access	Field Name (ID): Description
5:0	2h RW	**Read to Write DQ Delay Same Rank (TRWSR):** Specifies the delay from a DRAM Read to a Write command of the same rank (in DRAM clocks). • LPDDR3/LPDDR4/WIO2 Equation: tRWSR = RL + tDQSCKmax + BL/2 - WL + tWPRE. • DDR3L/DDR4 Equation: tRWSR = CL + tDQSCKmax + BL/2 - CWL + tWPRE. Note: For LPDDR3/4 using ODT, this latency may need to be increased by tODToffadj. Note: For DDR3L/DDR4 using ODT, this latency may need to be increased by one clock.

3.132 DRAM Timing Register 6A (D_CR_DTR6A) — Offset 1620h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1620h	20100000 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	20h RW	**Opportunistic Refresh Idle Timer (OREFDLY):** Rank idle period that defines an opportunity for refresh (in DRAM clocks).
23:19	2h RW	**Valid Clocks After CKE Low [tCKELCK/tCKELCS/tCPDED/tCKSRE] (TCKCKEL):** Specifies the amount of time that DRAM clocks need to toggle after CKE goes low (in DRAM Clocks). • For WIO2/LPDDR3, this covers tCPDED. • For LPDDR4, this covers both tCKELCK and tCKELCS. • For DDR3L/DDR4, this is tCKSRE. Note: D-Unit hardware enforces minimum of one SPID clocks after CKEL, any value in this register is the additional time.
18:15	0h RW	**Reserved (RSVD18_15):** Reserved
14:8	0h RW	**Mode Register Read to Any Command Delay (TPSTMRRBLK):** Specifies the quiet time after issuing MRR command (in DRAM Clocks). This is the channel block time for the MR19 command issued as part of LPDDR4 DQS Retraining flow. For all other MR commands this is the rank block time. Note: D-Unit treats MRR as a read and always applies relevant turnaround times, any value programmed in this CR must be greater than those turnaround times for D-Unit to enforce any additional time from MRR to the next read/write.
7	0h RO	**Reserved (RSVD7):** Reserved
6:0	0h RW	**Any Command to Mode Register Read/Write Delay (TPREMRBLK):** Specifies the channel quiet time before issuing MRR/MRW command. (in DRAM clocks). • LPDDR3 Equation: = tXP + tMRRI - 4. • LPDDR4 Equation: = tXP + tMRRI - 8. • DDR3L Equation: = CWL + BL/2 + 1 - 8 • DDR4 Equation: = CWL + BL/2 + 1 - 16. Note: D-Unit treats MRR as a read and MRW/MRS as a write and always applies relevant turnaround times, any value programmed in this CR must be greater than those turnaround times for D-Unit to enforce any additional time from previous read/writes.

3.133 DRAM Timing Register 7A (D_CR_DTR7A) — Offset 1624h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1624h	D060C06 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	3h RW	**All Bank Precharge to Activate Command Delay [tRPab] (TRPAB):** Specifies the delay between a DRAM Precharge All Bank command and a DRAM Activate command (in DRAM Clocks). Note: This CR should be constrained to a minimum of 4 in LP3 and minimum of 8 in LP4. Note: Derating adds 1.875ns to this timing. • For LPDDR, tRPpb = tRP, tRPab = tRP + 3ns. • For DDR3L, DDR4 and WIO2 8ch tRPpb = tRPab = tRP.
25:23	2h RW	**Mode Register Write to any Command Delay [tMRD/tMRW] (TPSTMRWBLK):** Specifies the quiet time after issuing MRW command (in 8 x DRAM clocks). Note: This time covers for both tMRD and tMRW.
22:16	6h RW	**Write Command to Power Down Delay [tWRPDEN] (TWRPDEN):** Specifies the minimum time between a write command to PowerDown command (in DRAM clocks). Must be at least equal to tWR + tCCD + tWL + 2.
15:9	6h RW	**Read Command to Power Down Delay [tRDPDEN] (TRDPDEN):** Specifies the minimum time between a read command to PowerDown command (in DRAM clocks). Must be at least equal to CL/RL + tDQSCKmax + tCCD + 1.
8:7	0h RO	**Reserved (RSVD8_7):** Reserved
6:0	6h RW	**Row Activation Period [tRAS] (TRAS):** Specifies the minimum delay between the DRAM Activate and Precharge commands to the same bank (in DRAM clocks). Note: Derating adds 1.875ns to this timing.

3.134 DRAM Timing Register 8A (D_CR_DTR8A) — Offset 1628h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1628h	CC50A18 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	3h RW	**Minimum Self-Refresh Time [tSR/tCKESR] (TCKESR):** Specifies the minimum time that DRAM should remain in SR (in DRAM clocks).
25:21	6h RW	**Minimum Low Power Mode Residency (LPMDRES):** Specifies the minimum time that PHY should remain in LPMode (in DRAM clocks).
20:15	Ah RW	**Low Power Mode Exit to Clock Enable Delay (LPMDTOCKEDLY):** Specifies the minimum time between the LP Mode exit to the CK stop/tristate deassertion and powerdown exit (in DRAM clocks). Note: Must be equal to t_idle_latency published in the DDRIO PHY and less than 0x3C.
14:8	Ah RW	**Clock Stop to Low Power Mode Delay (CKETOLPMDDLY):** Specifies the time between CK stop/tristate to the Low Power Mode entry. This timing parameter is used to delay Low Power Mode entry (in DRAM clocks). Note: Must be at least equal to t_idle_length parameter published in the DDRIO PHY and less than 0x7C.
7:0	18h RW	**Power Down Idle Timer (PWDDLY):** This is a non-JEDEC timing parameter used to delay powerdown entry (in DRAM clocks).

3.135 D-Unit ODT Control Register A (D_CR_DOCRA) — Offset 162Ch

Specifies the parameters to control DRAM ODT.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 162Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29	0h RW	**Rank 1 Read ODT Control (R1RDODTCTL):** Specifies the behavior of ODT signals when a Read command is issued to Rank 1. 0 - Read ODT is disabled for Rank 1 1 - Assert ODT to for Rank 0 (non-targeted Rank) Note: This register should be set to 0 for LPDDR3 devices
28	0h RW	**Rank 0 Read ODT Control (R0RDODTCTL):** Specifies the behavior of ODT signals when a Read command is issued to Rank 0. 0 - Read ODT is disabled for Rank 0 1 - Assert ODT to for Rank 1 (non-targeted Rank) Note: This register is reserved for LPDDR3 devices
27:26	0h RW	**Rank 1 Write ODT Control (R1WRODTCTL):** Specifies the behavior of ODT signals when a Write command is issued to Rank 1. 00 - Write ODT is disabled 01 - Assert ODT to Rank 0 (non-targeted Rank) 10 - Assert ODT to Rank 1 (targeted Rank) 11 - Assert ODT to Rank 0 and Rank 1 Note: 10 and 11 are reserved values for LPDDR3
25:24	0h RW	**Rank 0 Write ODT Control (R0WRODTCTL):** Specifies the behavior of ODT signals when a Write command is issued to Rank 0. 00 - Write ODT is disabled 01 - Assert ODT to Rank 0 (targeted Rank) 10 - Assert ODT to Rank 1 (non-targeted Rank) 11 - Assert ODT to Rank 0 and Rank 1 Note: 10 and 11 are reserved values for LPDDR3
23:18	0h RO	**Reserved (RSVD23_18):** Reserved
17:14	0h RW	**Read ODT assertion to de-assertion delay (DDR3L/DDR4) (RDODTSTOP):** Specifies Read ODT assertion to ODT de-assert delay (in DRAM clocks). DDR3L Equation: RDODTSTOP = 6. DDR4 Equation: RDODTSTOP = 5 + tRPRE. Note: Add 1 if DOCRx.RDODTSTART = CL - CWL in 2N mode.
13	0h RO	**Reserved (RSVD13):** Reserved
12:9	0h RW	**Read command to ODT assertion delay (DDR3L/DDR4) (RDODTSTART):** Specifies Read ODT assertion delay after Read Command (in DRAM clocks). DDR3L/DDR4 Equation: RDODTSTART = CL - CWL + tWPRE - tRPRE. Note: In DDR3L/DDR4 2N mode add 1 to enable termination at the start of read data burst.
8:5	0h RW	**Write ODT Assertion to De-assertion Delay (WRODTSTOP):** Specifies number of clocks after ODT assertion that D-Unit deasserts ODT signal (in DRAM clocks). LPDDR3 Equation: WRODTSTOP = RU(tODTon(max)/tCK) + RU((tDQSSmax+tWPST)/tCK) + BL/2 - RD(tODToffmin/tCK) DDR3L Equation: WRODTSTOP = 6. DDR4 Equation: WRODTSTOP = 5 + tWPRE. Note: Add 1 if DOCRx.WRODTSTART = 0 in 2N mode.
4	0h RO	**Reserved (RSVD4):** Reserved
3:0	0h RW	**Write command to ODT assertion delay (WRODTSTART):** Specifies number of clocks after Write command that D-Unit asserts ODT signal (in DRAM clocks). LPDDR3 Equation: WRODTSTART = WL - RU(tODTon(max)/tCK) DDR3L/DDR4 Equation: WRODTSTART = 0 Note: In DDR3L/DDR4 2N mode the value can be set to 0 to assert ODT one DRAM clock earlier than the Write Command (WR) or set to 1 to assert at the same clock as command (CS assertion).

3.136 D-Unit Power Management Control 0 (D_CR_DPMC0) — Offset 1630h

Specifies the parameters to control D-Unit power management features.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1630h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**Reserved (RSVD31_29):** Reserved
28:24	0h RW	**SUSPEND/SUSPENDP Power Management Message Opcode (SUSPMOP):** DDRIO PHY Power Mode Opcode: After the D-Unit has placed the DRAM devices in Self Refresh/PASR mode as the result of a SUSPEND/SUSPENDP message, it sends this 5-bit value to the DDRIO PHY to tell it which power saving mode it should enter. Changing this register value while in SUSPEND will have no effect. Note: This opcode cannot be a PM state where it disables PHY PLLs i.e PM7 in LPDDR PHY.
23	0h RO	**Reserved (RSVD23):** Reserved
22	0h RW	**PM Message Wait for Clock Gate Enable (SRPMCLKW):** Specifies when it is safe to send PM message to the PHY. When enabled, D-Unit waits for SPID Clock to deassert before sending a PM message on SR entry. • 0: D-Unit will not wait for SPID_clk to deassert before sending the PM message to PHY. • 1: D-Unit will wait for SPID_clk to deassert before sending PM message to the PHY. Note: The value must be 1 when DYNPMOP = 7h.
21:17	0h RW	**Dynamic Self-Refresh Power Management Message Opcode (DYNPMOP):** DDRIO PHY Power Mode Opcode: After the D-Unit has placed the DRAM devices in Self Refresh mode as the result of a Dynamic Self-Refresh, it sends this 5bit value to the DDRIO PHY to tell it which power saving mode it should enter. Changing this register value while in self-refresh will only change the PM state for the next entry in DynSR.
16	0h RW	**Dynamic Self-Refresh Enable (DYNSREN):** When set to 1, the D-Unit will automatically control DRAM Self Refresh entry and exit based on interface state and requests in pending queues. When there is no pending request in the queues and PMI is idle, then the D-Unit will place the DRAM devices in Self Refresh mode. The DRAM devices will be brought out of Self-Refresh when idle conditions don't hold.
15:0	0h RW	**Self-Refresh Entry Delay (SREDLY):** Specifies the minimum time the D-Unit will wait before it enters Dynamic Self-Refresh mode when idle (in 16x DRAM Clocks). Note: The value in this field needs to be minimum of 4 in functional mode and minimum of 50 in PSMI mode.

3.137 D-Unit Power Management Control 1 (D_CR_DPMC1) — Offset 1634h

Specifies the parameters to control D-Unit power management features.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1634h	10000028 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29	0h RW	**D-Unit Repeaters Clock Gate Disable (RPTCLKGTDIS):** Setting this bit to 0 allows majority of the repeaters between D-Unit and PHY to clock gate when there is no activity in order to save power. • 0 - Enable Repeaters clock gating. • 1 - Disable Repeaters clock gating. Note: This is a de-feature bit and should be set to 0 for normal operation. Note: The value should only change after DRAM Timing Registers (DTR) are programmed.
28	1h RW	**IOSF-SB End Point Clock Gate Disable (SBEPCLKGTDIS):** Setting this bit to 0 enables the clock gating of IOSF-SB End Points in D-Unit and CPGC when there is no IOSF-SB activity in order to save power. • 0 - Enable IOSF-SB EP clock gating. • 1 - Disable IOSF-SB clock gating. Note: This is a de-feature bit and should be set to 0 for normal operation. Note: The value should only change after DRAM Timing Registers (DTR) are programmed.
27	0h RW	**Local Clock Gate Disable (CLKGTDIS):** Setting this bit to 0 allows the majority of the D-Unit clocks to be gated off when there is no activity in order to save power. When set to 1, D-Unit clockgating is disabled. • 0: Enable. • 1: Disable. Note: This is a de-feature bit and should be set to 0 for normal operation. Note: The value should only change after DRAM Timing Registers (DTR) are programmed.
26	0h RW	**Chip Select Tristate Enable (CSTRIST):** • 0: The DRAM CS pins associated with the enabled ranks are never tristated. • 1: The DRAM CS pins are tristated when DRAM clock is stopped or tristated. Note: CS is not tristated when global tristate flow is disabled (DCBR.TRISTDIS = 1).
25:24	0h RW	**Command/Address Tristate (CMDTRIST):** • 00: The DRAM CA pins are never tristated. • 01: The DRAM CA pins are only tristated when all enabled CKE pins are low. • 10: The DRAM CA pins are tristated when not driving a valid command. • 11: Reserved
23:16	0h RW	**Partial Array Self-Refresh Segment Mask (PASR):** This is the Segment Mask used for the MRW to enable PASR during SUSPENDP (Partial Array Self Refresh entry).
15:8	0h RW	**Page Close Timeout Period (PCLSTO):** Specifies the time from the last access of a DRAM page until that page is scheduled to close by sending a Precharge command to DRAM (in 16 x DRAM clocks).
7	0h RW	**Page Close Timeout Disable (PCLSTODIS):** When disabled, D-Unit will not close the DRAM page when idle. • 0: Enable page close timer. • 1: Disable page close timer (Used during DRAM init and DDRIO training).
6	0h RO	**Reserved (RSVD6):** Reserved
5	1h RW	**ODT Tristate Enable (ODTTRIST):** • 0: The DRAM ODT pins associated with the enabled ranks are never tristated. • 1: DRAMs ODT pins are tristated when DRAM clock is stopped or tristated. Note: ODT is not tristated when global tristate flow is disabled (DCBR.TRISTDIS = 1)

Bit Range	Default & Access	Field Name (ID): Description
4:3	1h RW	**Clock Stop/Tristate Enable (ENCKSTP):** Enable/Disable CK Stop/Tristate During Power down. • 00: Disable CK Stop/Tristate During Power down. • 01: Enable CK Stop During Power down. • 10: Enable CK Tristate During Power down. • 11: Reserved Note: CK is not stopped or tristated when global tristate flow is disabled (DCBR.TRISTDIS = 1).
2:1	0h RW	**Low Power Mode Opcode (LPMODEOP):** D-Unit will send the value in this register after it has entered Powerdown Mode and has stopped/tristated the clock. 00: Disable LPMode. Note: LPMODE entry is not possible when global tristate flow is disabled (DCBR.TRISTDIS = 1).
0	0h RW	**Disable Power Down (DISPWRDN):** Setting this bit to 1 disables dynamic control of DRAM Power-Down entry and exit by keeping the CKE pins driven high. BIOS may set it to 1 during DRAM initialization and DDRIO training. This bit should be set to 0 for normal operation. • 0: The D-Unit dynamically controls the CKE pins to place the DRAM devices in Power Down mode and bring them out of Power Down mode. • 1: The D-Unit constantly drives the CKE pins high to keep the DRAM devices from entering Power Down mode when ranks are idle. Note: This bit is overridden if CKEMODE = 1. This bit does not control CKE behavior on SR entry/exit.

3.138 DRAM Refresh Control (D_CR_DRFC) — Offset 1638h

Specifies the parameters to control scheduling of refresh commands.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1638h	20001750 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	1h RW	**Maintenance Operation Channel Block Time (MNTCHNBLKTIME):** Specifies the amount of time that D-unit will block the channel before any DRAM maintenance operation is issued. (in SPIDclk)
28:25	0h RW	**Maintenance Operation Delay CAS Count (MNTDLYCASCOUNT):** When a critical read request is pending in RPQ and a maintenance operation (MRR, ZQCal, Ref, etc.) needs to be performed, D-unit delays the maintenance operation and allows this many read or write requests to be scheduled before allowing the maintenance operation. Note: This mode does not apply to Panic refreshes.
24:22	0h RO	**Reserved (RSVD24_22):** Reserved
21	0h RW	**Disable Refresh Debt Clear (DISREFDBTCLR):** When set, D-Unit will not clear refresh debt before Self Refresh SR Entry: • 0: D-Unit sends all postponed REF commands to DRAM before it enters Self Refresh. • 1: D-Unit enters SR without clearing the Refresh Debt (for Debug only).

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**Refresh Skew Disable (REFSKWDIS):** Disables Skewing of Refresh Counting between Ranks. Each rank has its own refresh counter. By default incrementing these refresh counters are skewed by 1/2 the tREFI period. Setting this bit to a 1 disables this feature and all refresh counters will increment at the same time per tREFI period. Skewing the tREFI counters can improve performance since traffic to all ranks does not have to be blocked to perform refresh. • 0: Incrementing the refresh counters are skewed by 1/2 tREFI period. • 1: All refresh counters will increment at the same time per tREFI period.
19:18	0h RO	**Reserved (RSVD19_18):** Reserved
17:16	0h RO	**Reserved (RSVD17_16):** Reserved
15	0h RW	**Extra Refresh Debit (EXTRAREFDBT):** When set to 1, D-Unit adds one extra refresh debit (for a total of two) on Self-refresh exit.
14:12	1h RW	**Minimum Refresh Rate (MINREFRATE):** Ensures that refresh rate never drops below a certain limit regardless of TQ polling. • 000: Stop issuing refresh commands and accumulating refresh debits. (does not stop tREFI counter). • 001: 0.25x refresh rate (i.e. 4x tREFI same as no limit). • 010: 0.5x refresh rate (i.e. 2x tREFI). • 011: 1x refresh rate (i.e. 1x tREFI). • 100: 2x refresh rate (i.e. 0.5x tREFI). • 101: 4x refresh rate (i.e. 0.25x tREFI). • 110: 4x refresh rate with derating forced on i.e. 0.25x tREFI. • 111: Reserved.
11:8	7h RW	**Refresh Panic Watermark (REFWMPNC):** When the Refresh counter per rank is greater than this value, the D-Unit will send a REF command to the rank regardless of pending requests. Note: REFWMPNC must be greater than or equal to REFWMHI and greater than 2, Max Value must be less than 8 to not violate 9xtREFI JEDEC requirement.
7:4	5h RW	**Refresh High Watermark (REFWMHI):** When the Refresh counter per rank is greater than this value, the D-Unit will send a REF command to the rank if there is no critical priority requests in the pending queues. Note: Value must be greater or equal to 1 and less than or equal to REFWMPNC.
3:1	0h RO	**Reserved (RSVD3_1):** Reserved
0	0h RW	**Opportunistic Refresh Disable (OREFDIS):** Disable opportunistic scheduling of refresh. • 0: D-Unit will send a REF command only if there is no pending request to that rank. • 1: D-Unit will not send any opportunistic refreshes. Refresh commands are only sent when the refresh counter is greater than REFWMHI. Note: When set, DISREFDBTCLR must also be set to be able to enter SR.

3.139 D-Unit Scheduler Control (D_CR_DSCH) — Offset 163Ch

Specifies parameters to control scheduling of commands to DRAM.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 163Ch	3901C08 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Early Write DQ Enable (EARLY_DQ_EN):** Enables D-unit to send the write data to the PHY one clock earlier than WL value (WL - 1). Note: Applicable to DDR3L and DDR4 only.
30:29	0h RW	**BGF Early Read Data Valid (BGF_EARLY_RDDATA_VALID):** Specifies the number of clocks the D-Unit sends the read data valid through the BGF earlier as compared to the data. • 00: Always write read valid in same SPID clock as data (STATIC_0). • 01: Always write read valid one SPID clock before data (STATIC_1). • 10: Write read valid up to 2 SPID clocks before data (DYNAMIC). • 11: Reserved
28:27	0h RW	**SPID Early Read Data Valid (SPID_EARLY_RDDATA_VALID):** Specifies the delay in SPID clocks from RDDATA_VALID assertion to actual data on SPID. The value should match what is programmed in DDRIO (PHY).
26:21	1Ch RW	**Write Pending Queue Count (WPQCOUNT):** Used to limit the number of available slots in Write Pending Queue/ Write Data Buffer. WPQCOUNT will only recognize changes when PMI ISM is not active.
20:16	10h RW	**Read Pending Queue Count (RPQCOUNT):** Used to limit the number of entries in Read Pending Queue. RPQCOUNT will only recognize changes when PMI ISM is not active.
15	0h RO	**Reserved (RSVD15):** Reserved
14:10	7h RW	**Read Return Data Additional Credits (BLKRDBF_ADD_RDDATA_CR):** Number of additional full cacheline (64B) read data return credits exposed to D-Unit when BLKRDBF is set. Note: The value in this field has no effect on Read return credits when BLKRDBF is not set.
9:8	0h RW	**In-Order Mode (INORDERMODE):** • 0h: In order mode disabled: Commands are sent out of order. • 1h: Partial in order mode: Read and Write CAS commands are sent in the order they were received. ACT and PRE can go out of order. • 2h: Full in order mode serialized test: All DRAM commands CAS ACT PRE associated with a PMI request are issued to DDR before any DRAM commands for a subsequent PMI request. • 3h: Reserved. In order modes should be enabled during init/training/CPGC testing. Should never be changed while the D-Unit queues are nonempty. For WIO2, when in-order mode is enabled (01 or 10), D_CR_DSCH_BYPASSEN must be set to 0
7	0h RW	**Idle Bypass Mode Enable (BYPASSEN):** When set new page hit/empty read requests will bypass D-Unit pipeline stages to save latency 0 - Disable Idle Bypass 1 - Enable Idle Bypass Note: Only applies to WIO2, this bit is reserved for other technologies
6	0h RW	**Block When RDB Full (BLKRDBF):** When set D-Unit stops scheduling new read commands to DRAM when the read data buffer (RDB) is full.
5:4	0h RW	**Stretch Mode (STRETCHMODE):** When stretch mode is enabled, commands are initiated only on Phase 0 of SPIDClk. • 00: Stretch mode is disabled. • 01: Commands are initiated on Phase 0 of every SPID clocks. • 10: Commands are initiated on Phase 0 of even SPID clocks. • 11: Commands are initiated on Phase 0 of odd SPID clocks.
3:0	8h RW	**Masked Write Turnaround Delta (TMWR_TA_DELTA):** The value in this register is subtracted from Masked Write to Read, Masked Write to Write and Masked Write to Masked Write turnaround times to account for half BL MWr commands in LPDDR4 and WIO2. • LPDDR4: = MWr tCCD = MWr BL/2 = 8. • WIO2: = NA.

3.140 DRAM Calibration Control (D_CR_DCAL) — Offset 1640h

Specifies parameters to control ZQ Calibration.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1640h	1057 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**ZQ Calibration Type (ZQCALTYPE):** Determines whether the ZQ Calibration is a long or short calibration command (due to ZQCALSTRT). 0: Short calibration (ZQCS). 1: Long calibration (ZQCL).
30	0h RW/V	**ZQ Calibration Start Rank 1 (ZQCALSTRTR1):** Set this bit to 1 to start the ZQ calibration sequence on Rank 1. This bit will remain a 1 until the ZQ calibration is complete for rank 1, then it will return to 0. 0: ZQ calibration is done. 1: ZQ calibration has started and is in progress.
29	0h RW/V	**ZQ Calibration Start Rank 0 (ZQCALSTRTR0):** Set this bit to 1 to start the ZQ calibration sequence on Rank 0. This bit will remain a 1 until the ZQ calibration is complete for rank 0, then it will return to 0. 0: ZQ calibration is done. 1: ZQ calibration has started and is in progress.
28:23	0h RO	**Reserved (RSVD28_23):** Reserved
22:21	0h RW	**Self-Refresh Exit ZQ Calibration Control (SRXZQC):** • 00: On DynSR exit ZQ timer determines the ZQ type. When the state is lost (i.e due to AutoPG/S0ix) ZQCL is always performed. • 01: Always perform ZQCL after self refresh exit. In LPDDR4, ZQ with traffic blocked. • 10: Always perform ZQCS on SR exit. For LPDDR4, ZQ while traffic is allowed. • 11: No ZQCL commands are sent (it disables ZQCAL commands on SR exit).
20:18	0h RO	**Reserved (RSVD20_18):** Reserved
17	0h RW	**ZQ Calibration Mode (ZQCLMODE):** Specifies how ZQCal commands are sent to different ranks. • 0: ZQCal commands are sent in parallel to all ranks. • 1: ZQCal commands are sent serially to each rank.
16	0h RW	**Periodic ZQ Calibration Disable (ZQCDIS):** • 0: Periodic ZQ Calibration is Enabled. • 1: Disable periodic ZQ Calibration.
15:14	0h RO	**Reserved (RSVD15_14):** Reserved
13:0	1057h RW	**ZQ Calibration Interval (ZQINT):** Specifies the time interval between two ZQCS (LPDDR3/DDR3L/DDR4) or ZQ Start (LPDDR4) commands to a DRAM device. (in RTC 32.8KHz clocks)

3.141 DRAM Mode Registers Shadow Copy (D_CR_MR_SHADOW) — Offset 1648h

This register contains a copy of Mode Regsiters in the DRAM so that D-unit can only modify certain bits without affecting other values.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1648h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:16	0h RW	**MR Value 2 (MR_VALUE2):** MR3 Shadow Register (DDR4): BIOS writes the correct value of MR3 register in DDR4 into this field at boot time. D-Unit modifies one bit and rewrites the MR3 into DDR4 to enter and exit MPR mode.
15:14	0h RO	**Reserved (RSVD15_14):** Reserved
13:0	0h RW	**MR Value (MR_VALUE):** MR3 Shadow Register (WIO2): BIOS sets the value of this field at boot time based on the DRAM device configuration. D-Unit merges the value in MR3_THERM_OFFSET with this field and writes the result into DRAM MR3 MR2 Shadow Register (DDR3L): BIOS writes the correct value of MR2 register in DDR3L into this field at boot time. D-Unit modifies one bit and rewrites the MR2 into DDR3L DRAM before SR entry. MR4 Shadow Register (DDR4): BIOS writes the correct value of MR4 register in DDR4 into this field at boot time. D-Unit modifies one bit and rewrites the MR4 into DDR4 DRAM after temperature read out.

3.142 VNN Scaling Timer Control (D_CR_VNNTIMER) — Offset 164Ch

Specifies parameters for VNN Scaling Timer in D-Unit. The values in this register will be set by P-code during VNN scaling period.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 164Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**VNN Scaling Timer Enable (VNN_TIMER_EN):** • 0: The D-Unit VNN Scaling Timer is disabled. • 1: The D-Unit VNN Scaling Timer is enabled.
30:12	0h RO	**Reserved (RSVD30_12):** Reserved
11:0	0h RW	**VNN Timer Time (VNN_TIMER_TIME):** The final timer value (in 16 x DRAM clocks).

3.143 Periodic DRAM Temperature Polling Control (TQ) (D_CR_TQCTL) — Offset 1650h

Specifies the control for periodic temperature monitoring and control of DRAM device.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1650h	6C000008 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	3h RW/V	**TQ Data Rank 1 (TQDATAR1):** If Rank 1 is disabled, this value will remain zero. This field contains the data of the last temperature sensor read from Rank 1 DRAM Mode Register. It is overwritten with each command.
28:26	3h RW/V	**TQ Data Rank 0 (TQDATAR0):** This field contains the data of the last temperature sensor read from Rank 0 DRAM Mode Register. It is overwritten with each command.
25:22	0h RO	**Reserved (RSVD25_22):** Reserved
21:8	0h RW	**TQ Poll Period (TQPOLLPER):** This sets the frequency by which the D-Unit initiates temperature sensor read from DRAM mode register to determine required refresh rate (in 4x tREFI units).
7	0h RW	**Temperature Controlled Refresh Range Enable (DDR4 Only) (TCRREN):** When set, after a DRAM temperature read out (MPR sequence), D-unit writes a 1 to bit 2 of MR_SHADOW.MR_VALUE when temperature sensor read out for that rank indicates a value higher than 0x1, and writes a 0 to that bit otherwise. The new MR_VALUE is then written into MR4 of DDR4 for each enabled rank.
6	0h RW/V	**TQ Poll Start Rank 1 (TQPOLL_START_R1):** When set Dunit will initiate temperature sensor read for Rank 1. Hardware will clear the bit once the MR read command is issued for this rank. Note: TQPOLLEN mus be 0 before this bit is set.
5	0h RW/V	**TQ Poll Start Rank 0 (TQPOLL_START_R0):** When set Dunit will initiate temperature sensor read for Rank 0. Hardware will clear the bit once the MR read command is issued for this rank. Note: TQPOLLEN mus be 0 before this bit is set.
4	0h RW	**Self Refresh Temperature Range Enable (DDR3 Only) (SRTEN):** When set, before every Self refresh entry, D-Unit writes a 1 to bit 7 of MR_SHADOW.MR_VALUE when TQDATA for that rank indicates a value higher then 0x3, and writes a 0 to that bit otherwise. The new MR_VALUE is then written into MR2 of DDR3 for each enabled rank.
3	1h RW	**Enable Dynamic Timing Derating (ENDERATE):** When set to 1, the Dynamic Timing Derating is enabled. When the D-Unit determines (via TQ polling) that the DRAM requires timing derating in addition to refresh interval adjustment, the D-Unit will automatically adjust the relevant timing parameters.
2	0h RW	**Enable TQ Data Push (TQDATAPUSHEN):** When set to 1, D-Unit pushes the data form the last temperature sensor read to a punit register.
1	0h RW	**Enable TQ Poll on Self-Refresh Exit (TQPOLLSREN):** This bit enables temperature sensor read on Self Refresh Exit. If disabled, D-Unit will not initiate MR Read sequence to read temperature value on Self-Refresh exit.
0	0h RW	**Enable Periodic TQ Poll (TQPOLLEN):** This bit enables periodic TQ Poll. If disabled, D-Unit will not read the DRAM's temperature sensor value periodically.

3.144 Temperature Offset Control (D_CR_TQOFFSET) — Offset 1654h

Specifies temperature offset and refresh rate adjustments requested by software.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1654h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD31_16):** Reserved
15:11	0h RO	**Reserved (RSVD15_11):** Reserved
10:8	0h RW	**MR4 Adder (MR4_ADDER):** D-Unit adds the value of this field to TQDATA read from MR4 the resulting value is used to control refresh rate and AC timing derating.
7:3	0h RO	**Reserved (RSVD7_3):** Reserved
2	0h RW/V	**MR3 Offset Update (MR3_OFFSET_UPDATE):** When set, D-Unit writes the merged value of MR3_VALUE and MR3_THERM_OFFSET into MR3 of DRAM. D-Unit clears this bit once the value is written.
1:0	0h RW	**MR3 Thermal Offset (MR3_THERM_OFFSET):** (WIO2 only) Everytime MR3_OFFSET_UPDATE is set, D-Unit merges the value in this field with MR3_VALUE and writes the result to MR3 in DRAM

3.145 D-Unit Control Operations (D_CR_DCO) — Offset 1658h

Specifies D-Unit initialization and control operation.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1658h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/V	**Initialization Complete (IC):** Indicates that initialization of the D-Unit has been completed. Memory accesses are permitted and maintenance operation begins. Until this bit is set to a 1, the memory controller will not accept DRAM requests from the Bunit/GSA/2LM (PMI ISMs will not leave idle). Note: Set this bit to 1 only when all other D-Unit registers have been configured. Usually set at the last configuration step by BIOS on cold/warm reset. D-Unit hardware sets this bit on SR exit.
30	0h RO/V	**DDRIO PHY Initialization Complete (DIOIC):** Status indication that the DDRIO PHY initialization is complete reflects the status spid_init_complete signal.
29	0h RO	**Reserved (RSVD29):** Reserved
28	0h RW	**PMI Control Select (PMICTL):** • 0: D-Unit PMI is connected to Bunit/GSA/2LM. • 1: D-Unit PMI is connected to CPGC. Note: D_CR_DSCH_BYPASSEN must be set to 0 in CPGC mode. Note: PMI must be idle and D-Unit BGF_RUN = 0 before changing the value in this register.
27:8	0h RO	**Reserved (RSVD27_8):** Reserved
7:4	0h RO	**Reserved (RSVD7_4):** Reserved
3	0h RW	**Enable PSMI Mode (PSMIEN):** When enabled, D-Unit will synchronize clock crossing signals. • 0: PSMI Mode is disabled. • 1: PSMI Mode is enabled. Note: Change only allowed when D-Unit is idle.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Maintenance Reset (MNTRST):** Writing a 1 to this field resets all maintenance timers. Clears all states and also clears refresh debt queues. This bit must be cleared by software after at least 3 SPID clocks.
1	0h RW	**Enable Maintenance Operations (MNTEN):** Setting this field to 1 enables all maintenance operations. When DCO.IC is set, the maintenance operations are enabled irrespective of the value of this field.
0	0h RO	**Reserved (RSVD0):** Reserved

3.146 Data Scrambler (D_CR_SCRAMCTRL) — Offset 16A4h

Specifies parameters to control data scrambling in D-Unit.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 16A4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Enable Data Scrambler (SCRM_EN):** When set to 1, data scrambling is enabled. When set to 0, data scrambling is disabled. Should be set before D_CR_BGF_CTL_BGF_RUN is set to 1.
30	0h RO	**Reserved (RSVD30):** Reserved
29:28	0h RW	**Scrambler Clock Gate Select (CLOCKGATE):** This field controls how the scrambler output code is clock gated to reduce power. • 00: Clock gate disabled. • 01: Clock Gate every 2 cycles. • 10: Clock Gate every 3 cycles. • 11: Clock Gate every 4 cycles.
27:16	0h RO	**Reserved (RSVD27_16):** Reserved
15:0	0h RW	**Scrambling Key (KEY):** Sets the key for the scrambler. The key should be a random value that is set following each cold boot.

3.147 Error Injection Address Register (D_CR_ERR_INJ) — Offset 16ACh

Contains the target address for ECC error injection.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 16ACh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD31):** Reserved
30:1	0h RW	**Error Injection Target Address (ADDRESS):** Specifies the PMI address of the write transaction to be injected with the error. Only applicable to Write transactions. Read/ under-fill read of the partial write operation is not affected.
0	0h RO	**Reserved (RSVD0):** Reserved

3.148 Error Injection Control Register (D_CR_ERR_INJ_CTL) — Offset 16B0h

Controls injecting correctable or uncorrectable errors into the write requests specified by target address.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 16B0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RSVD31_4):** Reserved
3	0h RW	**Error Injection Type Higher 32B (SEL_HI):** 0 - Uncorrectable Error (UE) is armed for write address matching to inject UE by using the same poisoning scheme, i.e. inverting corresponding write ECC[6:0] on every QW of the 32B data. 1 - Correctable Error (CE) is armed for write address matching to inject CE by inverting corresponding write ECC[0] on every QW of the 32B data.
2	0h RW	**Error Injection Enable Higher 32B (EN_HI):** When set the error injection is continuously armed for higher 32B of D_CR_ERR_INJ_ADDR write address matching until it is cleared.
1	0h RW	**Error Injection Type Lower 32B (SEL_LO):** 0 - Uncorrectable Error (UE) is armed for write address matching to inject UE by using the same poisoning scheme, i.e. inverting corresponding write ECC[6:0] on every QW of the 32B data. 1 - Correctable Error (CE) is armed for write address matching to inject CE by inverting corresponding write ECC[0] on every QW of the 32B data.
0	0h RW	**Error Injection Enable Lower 32B (EN_LO):** When set, the error injection is continuously armed for lower 32B of D_CR_ERR_INJ_ADDR write address matching until it is cleared.

3.149 Error Log Register (D_CR_ERR_ECC_LOG) — Offset 16B4h

Detected ECC errors are captured in this register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 16B4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/V	**CLEAR:** Setting this bit to one clears all fields in this register, including itself.
30:29	0h RW	**PMI VISA Byte Select (ECC_VISA):** Select ECC or PMI byte on VISA : • 00: ECC byte, • 01: PMI Data Byte [7:0], • 10: PMI Data Byte [63:56], • 11: PMI Data Byte [255:248]
28	0h RW/V	**Correctable Single-bit Error (CERR):** This bit is set when a correctable single-bit error occurs on a memory read data transfer. When this bit is set, the address that caused the error and the error syndrome are also logged and they are locked to further single bit errors, until this bit is cleared. A multiple bit error that occurs after this bit is set will override the address/error syndrome information.
27	0h RW/V	**Uncorrectable Multiple-bit Error (MERR):** This bit is set when an uncorrectable multiple-bit error occurs on a memory read data transfer. When this bit is set, the address that caused the error and the error syndrome are also logged and they are locked until this bit is cleared.
26:25	0h RW/V	**Error Burst Number (ERR_BURST):** Burst number (in BL8) of the error within a chunk.
24	0h RW/V	**Error Chunk Number (ERR_CHUNK):** Chunk number of the error. 0 - lower 32B chunk has error if MERR/CERR is set 1 - higher 32B chunk has the error if MERR/CERR is set
23:16	0h RW/V	**Quad Word ECC Syndrome (SYNDROME_QW):** ECC Syndrome for a QW (64 bit) within 32B Address
15:0	0h RW/V	**Request Tag (TAG):** Read Return Tag matches with the PMI Request Tag which triggered the error log.

3.150 D-Unit Fuse Status (D_CR_DFUSESTAT) — Offset 16BCh

Contains the values read from D-Unit fuses.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 16BCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD31_16):** Reserved
15:0	0h RO/V	**D-Unit Fuse Status (FUSESTAT):** D-Unit fuse bits are captured into this register and are available to be read. • [0]: fus_dun_ecc_dis. • [3:1]: fus_dun_max_supported_device_size[2:0]. • [4:4]: fus_dun_lpddr3_dis. • [5:5]: fus_dun_lpddr4_dis. • [6:6]: fus_dun_wio2_dis. • [7:7]: fus_dun_ddr3l_dis. • [8:8]: fus_dun_ddr4_dis. • [15:9]: reserved.

3.151 Major Mode Control (D_CR_MMC) — Offset 1724h

Specifies parameters to control read/write major mode operation and transitions.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1724h	2B01A518 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:27	5h RW	**RAW Conflict Read Priority for WMM Transition (RAW_WMM):** If a conflict read reaches this priority (or greater depending on access class occupancy), WMM will be triggered to unblock the corresponding write. D-Unit will stay in WMM until corresponding write is issued. Note: The value in this bit must not be higher than lowest terminal priority level of each access class.
26	0h RO	**Reserved (RSVD26):** Reserved
25:23	6h RW	**Read Isoch Trigger Priority (RIMPRIO):** If any read in the RPQ is at this programmable priority, RIM is triggered.
22:18	0h RO	**Reserved (RSVD22_18):** Reserved
17:12	1Ah RW	**Write Isoch Threshold (WIMTHRS):** When the number of entries in WPQ is greater than or equal to this value (higher than WMM entry watermark, less than WPQ size), it triggers write isoch mode (WIM).
11:6	14h RW	**Write Major Mode Exit Watermark (WMMEXIT):** When the number of entries in WPQ is less than this value, the D-Unit will switch back to read major mode.
5:0	18h RW	**Write Major Mode Entry Watermark (WMMENTRY):** When the number of entries in WPQ is greater than or equal to this value, the D-Unit will switch to write major mode (WMM). Note: the value must not be set to 0.

3.152 Major Mode RD/WR Counter (Set A and B) (D_CR_MMRDWR_AB) — Offset 1728h

Minimum read and maximum write counter control. This register defines the minimum number of reads in RMM and maximum number of writes in WMM before a mode transition happens.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1728h	1020408 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (RSVD31_26):** Reserved
25:20	10h RW	**Max Writes B (MAXWRB):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set B).

Bit Range	Default & Access	Field Name (ID): Description
19:14	8h RW	**Min Reads B (MINRDB):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set B).
13:12	0h RO	**Reserved (RSVD13_12):** Reserved
11:6	10h RW	**Max Writes A (MAXWRA):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set A).
5:0	8h RW	**Min Reads A (MINRDA):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set A).

3.153 Major Mode RD/WR Counter (Set C and D) (D_CR_MMRDWR_CD) — Offset 172Ch

Minimum read and maximum write counter control. This register defines the minumum number of reads in RMM and maximum number of writes in WMM before a mode transition happens (sets C and D).

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 172Ch	1020408 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (RSVD31_26):** Reserved
25:20	10h RW	**Max Writes D (MAXWRD):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set D).
19:14	8h RW	**Min Reads D (MINRDD):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set D).
13:12	0h RO	**Reserved (RSVD13_12):** Reserved
11:6	10h RW	**Max Writes C (MAXWRC):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set C).
5:0	8h RW	**Min Reads C (MINRDC):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set C).

3.154 Access Class Initial Priority (D_CR_ACCIP) — Offset 1730h

Each field of this register defines the initial priority of one access class.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1730h	17C2 h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD31_15):** Reserved
14:12	1h RW	**Access Class 4 Initial Priority (AC4IP):** Initial priority level of read requests coming with access class 4.
11:9	3h RW	**Access Class 3 Initial Priority (AC3IP):** Initial priority level of read requests coming with access class 3.
8:6	7h RW	**Access Class 2 Initial Priority (AC2IP):** Initial priority level of read requests coming with access class 2.
5:3	0h RW	**Access Class 1 Initial Priority (AC1IP):** Initial priority level of read requests coming with access class 1.
2:0	2h RW	**Access Class 0 Initial Priority (AC0IP):** Initial priority level of read requests coming with access class 0.

3.155 Access Class 0 Priority Promotion Control (D_CR_RD_PROM0) — Offset 1734h

This register defines the priority promotion policy for access class 0. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1734h	1F52940 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
24:20	1Fh RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	Ah RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	Ah RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	Ah RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.156 Access Class 1 Priority Promotion Control (D_CR_RD_PROM1) — Offset 1738h

This register defines the priority promotion policy for access class 1. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the associated level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1738h	14000000 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	Ah RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
24:20	0h RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	0h RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	0h RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	0h RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.157 Access Class 2 Priority Promotion Control (D_CR_RD_PROM2) — Offset 173Ch

This register defines the priority promotion policy for access class 2. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 173Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

Bit Range	Default & Access	Field Name (ID): Description
24:20	0h RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	0h RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	0h RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	0h RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.158 Access Class 3 Priority Promotion Control (D_CR_RD_PROM3) — Offset 1740h

This register defines the aging policy for access class 3. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1740h	1F29400 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
24:20	1Fh RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	5h RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	5h RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	0h RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.159 Access Class 4 Priority Promotion Control (D_CR_RD_PROM4) — Offset 1744h

This register defines the aging policy for access class 3. Each field of this register defines the number of CASes that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1744h	1F5294A h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
24:20	1Fh RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	Ah RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	Ah RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	Ah RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	Ah RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.160 Deadline Threshold (D_CR_DL_THRS) — Offset 1748h

Specifies when the request with initial priority 0 get promoted to a higher priority level.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1748h	6 h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD31_11):** Reserved
10:0	6h RW	**Deadline Threshold (DEADLINE_THRS):** A requests with initial priority of 0 will exit priority 0 when its deadline is equal or less than this value plus current time. This field does not affect the priority of any requests in access classes with initial priority bigger than 0.

3.161 Major Mode Blocking Rules Control (D_CR_MM_BLK) — Offset 174Ch

This register controls blocking rules enforced in RMM and WMM.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 174Ch	1800 h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD31_25):** Reserved
24	0h RW	**WMM Regular Rule 1 (WMM_REG_R1):** Disable WMM unsafe write page hits block safe write page misses same bank.
23:20	0h RO	**Reserved (RSVD23_20):** Reserved
19	0h RW	**WMM Priority Rule 4 (WMM_PRIO_R4):** Disable WMM unsafe priority 1 read miss block write hit to same bank. Note: This rule does not block the bank that is being blocked by WMM_PRIO_R3. Priority rules 1, 3 and 4 should be enabled/disabled together.
18	0h RW	**WMM Priority Rule 3 (WMM_PRIO_R3):** Disable WMM unsafe priority 1 write hit block write miss to same bank. Note: This rule does not block the bank that is being blocked by WMM_PRIO_R1. Priority rules 1,3 and 4 should be enabled/disabled together.
17	0h RW	**WMM Priority Rule 2 (WMM_PRIO_R2):** Disable WMM CAS block rule.
16	0h RW	**WMM Priority Rule 1 (WMM_PRIO_R1):** Disable WMM unsafe top priority 1 write miss block write hit same bank. Priority rules 1, 3 and 4 should be enabled/disabled together.
15:14	0h RO	**Reserved (RSVD15_14):** Reserved
13	0h RW	**RMM Regular Rule 6 (RMM_REG_R6):** Disable RMM unsafe write page hits block safe write page misses same bank.
12	1h RW	**RMM Regular Rule 5 (RMM_REG_R5):** Disable RMM unsafe read page miss block all safe and unsafe write page hit to the same bank. Note: This field must not be set to 0 (enabled) if RMM_REG_R4 is also 0.
11	1h RW	**RMM Regular Rule 4 (RMM_REG_R4):** Disable RMM unsafe write page hit block safe read page miss same bank. Note: This field must not be set to 0 (enabled) if RMM_REG_R5 is also 0.
10	0h RW	**RMM Regular Rule 3 (RMM_REG_R3):** Disable RMM unsafe read page hit block safe read and write page miss same bank. Note: This rule does not block the bank that is being blocked by RMM_PRIO_R3 and RMM_PRIO_R1.
9	0h RW	**RMM Regular Rule 2 (RMM_REG_R2):** Disable RMM unsafe read page empty block safe write page empty same rank.
8	0h RW	**RMM Regular Rule 1 (RMM_REG_R1):** Disable RMM unsafe read page hit block safe write page hit same rank.
7:4	0h RO	**Reserved (RSVD7_4):** Reserved
3	0h RW	**RMM Priority Rule 4 (RMM_PRIO_R4):** Disable RMM unsafe critical read miss block read and write hit to same bank. Note: This rule does not block the bank that is being blocked by RMM_PRIO_R3. Priority rules 1, 3 and 4 should be enabled/disabled together.
2	0h RW	**RMM Priority Rule 3 (RMM_PRIO_R3):** Disable RMM unsafe critical read hit block read and write miss to same bank. Note: This rule does not block the bank that is being blocked by RMM_PRIO_R1. Priority rules 1, 3 and 4 should be enabled/disabled together.
1	0h RW	**RMM Priority Rule 2 (RMM_PRIO_R2):** Disable RMM CAS block rule.
0	0h RW	**RMM Priority Rule 1 (RMM_PRIO_R1):** Disable RMM unsafe top critical read miss block read and write hit same bank. Note: Priority rules 1, 3 and 4 should be enabled/disabled together.

3.162 DRAM Self-Refresh Command (D_CR_DRAM_SR_CMD) — Offset 1754h

Self refresh command register to allow sending WAKE and SUSPEND messages to D-Unit. (Only one bit can be set at a time). Posted writes to this register are not completed until hardware clears the field.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1754h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RSVD31_4):** Reserved
3	0h RW/V	**SUSPENDP:** A SUSPENDP message will put the DRAM into self-refresh mode. The D-Unit will complete servicing outstanding memory requests and flush all queued Refresh commands to DRAM before putting the DRAM in self refresh mode. Finally, a PM message will be sent to the PHY. The bit is cleared by hardware after the PHY indicates the transition requested in the PM message has been completed. D-Unit will perform an MRW to MR17 with an opcode as defined by DPMC0.PASR before it places the DRAM into Self-Refresh.
2	0h RW/V	**SUSPEND:** A SUSPEND message will put the DRAM into self-refresh mode. The D-Unit will complete servicing outstanding memory requests and flush all queued Refresh commands to DRAM before putting the DRAM in Self Refresh mode. Finally, a PM message will be sent to the PHY. The bit is cleared by hardware only after the PHY indicates the transition requested in the PM message has been completed. Note: When COLDWAKE is set prior of setting this bit the DRAM will not be placed in SR.
1	0h RO	**Reserved (RSVD1):** Reserved
0	0h RW/V	**WAKE:** Take PHY out of PM states and wakes the DRAM out of self refresh mode. The bit is cleared by hardware only when the DRAM has exited out of self refresh mode and is accessible. Note: When COLDWAKE is set prior of setting this bit the D-Unit will not send SR exit command and will not set the DCO.IC bit.

3.163 DQS Retraining Control (D_CR_DQS_RETRAINING_CTL) — Offset 1780h

LPDDR4 DQS Retraining control register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1780h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DQS Periodic Retraining Interval (DQS_RETRAIN_INT):** This sets the frequency by which the D-Unit initiates periodic retraining (in 1x NREFI).
15:14	0h RO	**Reserved (RSVD15_14):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:4	0h RW	**DQS Oscillator Runtime (DQS_OSC_RT):** After D-Unit starts DQS oscillator, it must wait this amount of time before being able to read the value in MR18 and MR19 (in 16x DRAM clocks). Value in this register must be at least equal to DRAM's MR23 value. + tOSCO.
3	0h RW/V	**DQS Retrain Start Rank 1 (DQS_RETRAIN_START_R1):** When set, Dunit will initiate LPDDR4 DQS training sequence on Rank1. Hardware will clear bit when sequence completes (after issuing MR19). Note: DQS_RETRAIN_EN must be 0 before setting this bit.
2	0h RW/V	**DQS Retrain Start Rank 0 (DQS_RETRAIN_START_R0):** When set, Dunit will initiate LPDDR4 DQS training sequence on Rank0. Hardware will clear bit when sequence completes (after issuing MR19). Note: DQS_RETRAIN_EN must be 0 before setting this bit.
1	0h RW	**DQS Retrain SRX Exit (DQS_RETRAIN_SRX_EN):** Enable retraining on SR exit. This bit enables LPDDR4 DQS retraining on Self Refresh Exit. If disabled, D-Unit will not perform retraining on SR exit.
0	0h RW	**DQS Retrain Enable (DQS_RETRAIN_EN):** Periodic retraining enable: This bit enables periodic DQS retraining. If disabled, D-Unit will not perform retraining periodically. Note: Will be enabled only if DCO.IC is set and refreshes are enabled in DRF.MINREFRATE.

3.164 MR4 De-Swizzle Control (D_CR_MR4_DESWIZZLE) — Offset 1784h

Controls the data bits swizzling crossbar for MR4.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1784h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD31):** Reserved
30:28	0h RW	**MR4 Bit 2 Select 2nd Byte (MR4_BIT2_SEL2):** Selects bit 2 of MR4 data.
27	0h RO	**Reserved (RSVD27):** Reserved
26:24	0h RW	**MR4 Bit 1 Select 2nd Byte (MR4_BIT1_SEL2):** Selects bit 1 of MR4 data
23	0h RO	**Reserved (RSVD23):** Reserved
22:20	0h RW	**MR4 Bit 0 Select 2nd Byte (MR4_BIT0_SEL2):** Selects bit 0 of MR4 data.
19:18	0h RO	**Reserved (RSVD19_18):** Reserved
17:16	0h RW	**MR4 Byte 2 Select (MR4_BYTE_SEL2):** Selects byte position of the MR4 data for second device.
15	0h RO	**Reserved (RSVD15):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
14:12	0h RW	**MR4 Bit 2 Select (MR4_BIT2_SEL):** Selects bit 2 of MR4 data.
11	0h RO	**Reserved (RSVD11):** Reserved
10:8	0h RW	**MR4 Bit 1 Select (MR4_BIT1_SEL):** Selects bit 1 of MR4 data.
7	0h RO	**Reserved (RSVD7):** Reserved
6:4	0h RW	**MR4 Bit 0 Select (MR4_BIT0_SEL):** Selects bit 0 of MR4 data.
3:2	0h RO	**Reserved (RSVD3_2):** Reserved
1:0	0h RW	**MR4 Byte Select (MR4_BYTE_SEL):** Selects byte position of the MR4 data first device.

3.165 DRAM Rank Population 0 (D_CR_DRP0) — Offset 1A00h

Rank configuration register.

This is the start of multicast dunit registers. Writing to these will write to all active dunits.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A00h	10000000 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RW	**DRAM Device Per Rank (DRAMDEVICE_PR):** Specifies the number of DRAM devices that are ganged together to form a single rank. • 00: 1 DRAM device in each rank. • 01: 2 DRAM devices in each rank. • 10: 4 DRAM devices in each rank. • 11: 8 DRAM devices in each rank. Note: The actual number of devices is one more than the value programmed when ECC is enabled.
29:28	1h RW	**Address Decode (ADDRDEC):** Specifies the address mapping to be used: • 00: 1KB (A). • 01: 2KB (B). • 10: 4KB (C). • 11: Reserved.
27:25	0h RW	**Burst Length Mode (BLMODE):** • 000: Fixed BL8. • 001: Onthefly BL8. • 010: Fixed BL16. • 011: Onthefly BL16. • 100: Fixed BL32. • 101: Onthefly BL32. • 110-111: Reserved.

Bit Range	Default & Access	Field Name (ID): Description
24:22	0h RW	**DRAM Type (DRAMTYPE):** • 000: Reserved. • 001: Reserved. • 010: LPDDR4. • 011: Reserved. • 100: DDR4. • 101-111: Reserved. Note: The D-Unit should only use this field if allowed by fuse.
21	0h RW	**ECC Enable (ECCEN):** • 0: ECC is disabled. • 1: ECC is enabled. This bit determines if the D-Unit treats the PMI BE_ECC bits as ECC bits or Byte Enables. The D-Unit should not allow this bit to be set if ECC is disabled by fuse. This should only be used in configurations that support ECC.
20:19	0h RW	**CA Swizzle Type (CASWIZZLE):** • 00: uniDIMM/SODIMM/UDIMM. • 01: BGA. • 10: NA. • 11: uniDIMM/SODIMM/UDIMM with Rank 1 mirrored (DDR4 Only).
18:17	0h RO	**Reserved (RSVD18_17):** Reserved
16	0h RW	**Fine Bank Group Interleaving Enable (FBGINTEN):** When enabled, D-unit will use a lower order address bit in the bank hashing function (DDR4 Only). • 0: Bank Group Interleave disabled. • 1: Bank Group Interleave enabled. Note: BAHEN must be set for this bit to take effect.
15	0h RW	**Bank Address Hashing Enable (BAHEN):** • 0: Bank Address Hashing disabled. • 1: Bank Address Hashing enabled.
14	0h RW	**Rank Select Interleave Enable (RSIEN):** • 0: Rank Select Interleaving disabled. • 1: Rank Select Interleaving enabled.
13:9	0h RO	**Reserved (RSVD13_9):** Reserved
8:6	0h RW	**DRAM Device Density (DDEN):** Density of the DRAM devices populated on Ranks 0 and 1. • 000: 4 Gb. • 001: 6 Gb. • 010: 8 Gb. • 011: 12 Gb. • 100: 16 Gb. • 101-111: Reserved. Note: For LPDDR4 this value is the die density.
5:4	0h RW	**DRAM Device Data Width (DWID):** Data width of the DRAM device populated on Ranks 0 and 1. • 00: x8. • 01: x16. • 10: x32. • 11: x64.
3	0h RO	**Reserved (RSVD3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Dual Data Mode Enable (DDMEN):** • 0: PMI Dual Data Mode is disabled in D-Unit. • 1: PMI Dual Data Mode is enabled. Note: Dual Data Mode must be enable for DDR3L/DDR4 configurations.
1	0h RW	**Rank Enable 1 (RKEN1):** Enable Rank 1: Must be set to 1 to enable use of this rank.
0	0h RW	**Rank Enable 0 (RKEN0):** Enable Rank 0: Must be set to 1 to enable use of this rank. Note: Setting this bit to 0 is not a functional mode.

3.166 DRAM Timing Register 9A (D_CR_DTR9A) — Offset 1A04h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A04h	14A546 h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RO	**Reserved (RSVD31_23):** Reserved
22:18	5h RW	**Write to Write DQ Delay Same Bank Group Same Rank (TWWSR_L):** Specifies the delay from a DRAM Write to another Write command within the same bank group of the same rank (in DRAM clocks). DDR4 Equation: tWWSR_L = tCCD_L.
17:13	5h RW	**Read to Read DQ Delay Same Bank Group Same Rank (TRRSR_L):** Specifies the delay from a DRAM Read to another Read command within the same bank group of the same rank (in DRAM clocks). DDR4 Equation: tRRSR_L = tCCD_L.
12:6	15h RW	**Write to Read DQ Delay Same Bank Group Same Rank (TWRSR_L):** Specifies the delay from a DRAM Read to Write command within the same bank group of the same rank (in DRAM clocks). • DDR4 Equation: tWRSR_L = CWL + tDQSSmax + BL/2 + tWPST + tWTR_L.
5:0	6h RW	**Row Activation to Row Activation to Same Bank Group Delay [tRRD_L] (TRRD_L):** Specifies the minimum delay (in DRAM clocks) between two DRAM Activate commands to different banks within the same bank group of a rank. (DDR4 Only)

3.167 DRAM Timing Register 0A (D_CR_DTR0A) — Offset 1A08h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A08h	820702CB h

Bit Range	Default & Access	Field Name (ID): Description
31:27	10h RW	**Valid Clocks Before CKE High [tCKCKEH/tCSCKEH/tCKSRX] (TCKCKEH):** Number of valid clocks before CKE high (in DRAM clocks). • LPDDR4: The value in this register covers both tCKCKEH and tCSCKEH. • DDR3L/DDR4/LPDDR3/WIO2: The value covers tCKSRX which is defined as the number of valid DRAM clocks that have to toggle before the issuing of the Self Refresh Exit SRX. This value is also used if the clock frequency is changed or the clock is stopped or tristated during Power Down i.e. the number valid DRAM clocks that have to toggle before the issuing of the Power Down Exit PDX command. TCKCKEH can be used to compensate for clock stabilization delays in the motherboard. Note: D-unit hardware enforces minimum of two SPID clock before CKEH, any value in this register is the additional time.
26:22	8h RW	**Exit Self-Refresh to Valid Commands Requiring a Locked DLL Delay [tXSDLL] (TXSDLL):** D-Unit waits max(tXSR+tZQCL/tZQCS, tXSDLL) before allowing traffic to DRAM (in 32 x DRAM Clocks). LPDDR3/LPDDR4/WIO2: tXSDLL = 0. DDR3L/DDR4: tXSDLL = tDLLK. Note: In the equation above, tZQCL/tZQCS = 0 if no ZQ is performed on SR exit.
21:12	70h RW	**Exit Self-Refresh to Valid Command Delay [tXS/tXSR] (TXSR):** DDR3L/DDR4: tXS - Delay between Self Refresh Exit SRX to any DRAM Command not requiring DLL Lock. LPDDR/WIO2: tXSR - Delay between Self Refresh Exit SRX to any DRAM Command. (in DRAM clocks).
11:6	Bh RW	**Activate RAS to CAS Command Delay [tRCD] (TRCD):** Specifies the delay between a DRAM Activate command and a DRAM Read or Write command to the same bank (in DRAM clocks). Note: Derating adds 1.875ns to this timing.
5:0	Bh RW	**Precharge to Activate Command Delay of a Single Bank [tRPpb] (TRPPB):** Specifies the delay between a DRAM Precharge command and a DRAM Activate command to the same bank (in DRAM Clocks). Note : this CR should be constrained to a minimum of 4 in LPDDR3/DDR3L/WIO2 and minimum of 8 in LPDDR4/DDR4. Note: Derating adds 1.875ns to this timing.

3.168 DRAM Timing Register 1A (D_CR_DTR1A) — Offset 1A0Ch

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A0Ch	30481218 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	6h RW	**Exit Power Down to Next Command Delay [tXP] (TXP):** Specifies the delay from the DRAM Power Down Exit (PDX) command to any valid command (in DRAM clocks). Note: The value in this field must be programmed to tXPDLL when Slow Exit Mode Power-down is enabled for DDR3L.
26	0h RO	**Reserved (RSVD26):** Reserved
25:14	120h RW	**ZQ (long) Calibration Time [tZQCL/tZQCAL] (TZQCL):** • LPDDR3/DDR3L/DDR4: tZQCL/tZQoper - Specifies the delay between the DRAM ZQ Calibration Long (ZQCL) command and any DRAM command during normal operation. • LPDDR4: tZQCAL - ZQ Calibration time (in DRAM clocks). Note: This field defines the ZQ Calibration Long delay during normal operation. It is not the same as tZQinit which uses the same ZQCL command but the delay is longer. tZQinit applies only during poweron initialization of the DRAM devices and tZQoper applies during normal operation. BIOS executes the DRAM initialization sequence so it has to ensure tZQinit is met and not the D-Unit.

Bit Range	Default & Access	Field Name (ID): Description
13:6	48h RW	**ZQ Short Calibration Time [tZQCS] (TZQCS):** ZQCS to any DRAM Command Delay: Specifies the delay between the DRAM ZQ Calibration Short (ZQCS) command and any DRAM command (in DRAM clocks). DDR3L/DDR4/LPDDR3 only. LPDDR4 does not support ZQCS command
5:0	18h RW	**ZQ Latch Time [tZQLAT] (TZQLAT):** Specifies the delay between the DRAM ZQ Calibration Latch command and any DRAM command (in DRAM clocks). LPDDR4 only. Not used in DDR3L/DDR4/LPDDR3/WIO2.

3.169 DRAM Timing Register 2A (D_CR_DTR2A) — Offset 1A10h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A10h	46080C30 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	118h RW	**All Bank Refresh Cycle Time [tRFCab] (NRFCAB):** Specifies the delay between the REFab command to the next valid command. (in DRAM clocks)
21	0h RO	**Reserved (RSVD21):** Reserved
20:17	4h RW	**CKE Minimum Pulse Width [tCKE] (TCKE):** Specifies the minimum time from CKEL to CKEH (in DRAM clocks).
16	0h RO	**Reserved (RSVD16):** Reserved
15:0	C30h RW	**Refresh Interval Time [tREFI] (NREFI):** Specifies the average time between refresh commands. JEDEC Base Refresh Interval time (in DRAM clocks). Note: D-Unit will ignore the 2 LSBs of this field.

3.170 DRAM Timing Register 3A (D_CR_DTR3A) — Offset 1A14h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A14h	3002EA28 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	6h RW	**Read to Precharge Delay [tRDPRE] (TRTP):** Specifies the minimum delay between the DRAM Read and Precharge commands to the same bank (in DRAM clocks). • LPDDR3 Equation: = BL/2 + tRTP - 4. • LPDDR4 Equation: = BL/2 + Max (8, tRTP) - 8. • WIO2 Equation: NA. • DDR3L/DDR4 Equation: = tRTP.

Bit Range	Default & Access	Field Name (ID): Description
26:20	0h RW	**CAS to CAS Command Delay Adder (TCCD_INC):** Specifies the number of clocks to be added to turnaround times (for Stretch Mode). It increases delay between Read to Read or Read to Write commands by the amount programmed in this field (in 4 x DRAM clocks).
19:13	17h RW	**Write to Precharge Command Delay [tWRPRE] (TWTP):** Specifies the minimum delay between the DRAM Write command and the Precharge command to the same bank (in DRAM clocks). • LPDDR3/LPDDR4/WIO2 Equation: tWTP = BL/2 + WL + tWR + 1. • DDR3L/DDR4 Equation: tWTP = BL/2 + CWL + tWR.
12:11	1h RW	**DRAM Command Valid Duration (TCMD):** Specifies the number of DRAM clocks a command is held valid on the DRAM Address and Control buses. 1N is the DDR3 basic requirement. 2N is the extended mode for board signal integrity. • 0h: Reserved. • 1h: 1 DRAM Clock (1N). • 2h: 2 DRAM Clocks (2N). • 3h: Reserved. Note: DDR3L/DDR4 only. tCMD must be set to 1N for LPDDR3/LPDDR4/WIO2.
10:6	8h RW	**Write Latency [WL/CWL] (TCWL):** The delay between the internal write command and the availability of the first word of DRAM input data (in DRAM clocks).
5:0	28h RW	**Write CAS to Masked Write CAS Delay Same Bank (TWMWSB):** Specifies the minimum delay between DRAM Write command to Masked Write command to same bank (in DRAM clocks). • LPDDR4 Equation: tWMWSB = tCCDMW (BL16) or tCCDMW + 8 (BL32). • WIO2 Equation: NA. Note: Masked Write operation in LPDDR4 is always BL16 and in WIO2 is always BL4. D-Unit applies this timing for same rank as well as same bank.

3.171 DRAM Timing Register 4A (D_CR_DTR4A) — Offset 1A18h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A18h	30209149 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	30h RW	**Four Bank Activate Window [tFAW] (TFAW):** A rolling timeframe in which a maximum of four Activate commands can be issued to the same rank. This is to limit the peak current draw from the DRAM devices (in DRAM clocks).
23:18	8h RW	**Write to Read DQ Delay Different Ranks (TWRDR):** Specifies the delay from the start of a Write data burst of one rank to the start of a Read data burst of a different rank (in DRAM clocks). • LPDDR3 Equation: tWRDR = WL + tDQSSmax + BL/2 - (RL + tDQSCKmin - tRPRE). • LPDDR4 Equation: tWRDR = WL - RL + BL/2 + 4 - tDQSCKmin. • DDR3L/DDR4 Equation: tWRDR = CWL + tDQSSmax + BL/2 - (CL + tDQSCKmin - tRPRE). Note: For LPDDR3/4 using ODT, this latency may need to be increased by tODToffadj.

Bit Range	Default & Access	Field Name (ID): Description
17:12	9h RW	**Read to Write DQ Delay Different Ranks (TRWDR):** Specifies the delay from the start of a Read data burst of one rank to the Start of a Write data burst of a different rank (in DRAM clocks). • LPDDR3/LPDDR4 Equation: tRWDR = RL + tDQSCKmax + BL/2 - (WL - tWPRE). • DDR3L/DDR4 Equation: tRWDR = CL + tDQSCKmax + BL/2 - (CWL - tWPRE). Note: For LPDDR3/4 using ODT, this latency may need to be adjusted by tODTon. Note: For DDR3L/DDR4 using ODT, this latency may need to be increased by one clock.
11:6	5h RW	**Write to Write DQ Delay Different Ranks (TWWDR):** Specifies the delay from the start of a Write data burst of one rank to the start of a Write data burst of a different rank (in DRAM clocks). • LPDDR3/DDR3L/DDR4 Equation: tWWDR = BL/2 + tDQSSmax - tDQSSmin + tWPRE. • LPDDR4 Equation: tWWDR = BL/2 + 4 - tDQSSmin. Note: For LPDDR3/4 using ODT, this latency may need to be increased by tODToffadj.
5:0	9h RW	**Read to Read DQ Delay Different Ranks (TRRDR):** Specifies the delay from the start of a Read data burst of one rank to the Start of a Read data burst of a different rank (in DRAM clocks). • Equation: tRRDR = BL/2 + tDQSCKmax - tDQSCKmin + tRPRE.

3.172 DRAM Timing Register 5A (D_CR_DTR5A) — Offset 1A1Ch

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A1Ch	304200C2 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	6h RW	**Row Activation to Row Activation Delay [tRRD] (TRRD):** Specifies the minimum delay in DRAM clocks between two DRAM Activate commands to the same rank but different banks (tRC is the minimum delay between activations of the same bank). Note: Derating adds 1.875ns to this timing.
26	0h RO	**Reserved (RSVD26):** Reserved
25:23	0h RW	**Derate Increment (TDERATE_INC):** Specifies the additional delay that is added to DRAM timing when indicated by MR4 status. (in DRAM clocks) LPDDR3/LPDDR4/WIO2: Value is 1.875ns. Note: The value in this register is only added to these timing parameters: tRCD, tRAS, tRP and tRRD.
22:18	10h RW	**Write to Write DQ Delay Same Rank (TWWSR):** Specifies the delay from a DRAM Write to another Write command of the same rank (in DRAM clocks). Equation: tWWSR = tCCD.
17:13	10h RW	**Read to Read DQ Delay Same Rank (TRRSR):** Specifies the delay from a DRAM Read to another Read command of the same rank (in DRAM clocks). Equation: tRRSR = tCCD.
12:6	3h RW	**Write to Read DQ Delay Same Rank (TWRSR):** Specifies the delay from a DRAM Read to Write command of the same rank (in DRAM clocks). • LPDDR3/LPDDR4/WIO2 Equation: tWRSR = WL + tDQSSmax + BL/2 + tWTR. • DDR3L Equation: tWRSR = CWL + tDQSSmax + BL/2 + tWTR. • DDR4 Equation: tWRSR = CWL + tDQSSmax + BL/2 + tWTR_S.

Bit Range	Default & Access	Field Name (ID): Description
5:0	2h RW	**Read to Write DQ Delay Same Rank (TRWSR):** Specifies the delay from a DRAM Read to a Write command of the same rank (in DRAM clocks). • LPDDR3/LPDDR4/WIO2 Equation: tRWSR = RL + tDQSCKmax + BL/2 - WL + tWPRE. • DDR3L/DDR4 Equation: tRWSR = CL + tDQSCKmax + BL/2 - CWL + tWPRE. Note: For LPDDR3/4 using ODT, this latency may need to be increased by tODToffadj. Note: For DDR3L/DDR4 using ODT, this latency may need to be increased by one clock.

3.173 DRAM Timing Register 6A (D_CR_DTR6A) — Offset 1A20h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A20h	20100000 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	20h RW	**Opportunistic Refresh Idle Timer (OREFDLY):** Rank idle period that defines an opportunity for refresh (in DRAM clocks).
23:19	2h RW	**Valid Clocks After CKE Low [tCKELCK/tCKELCS/tCPDED/tCKSRE] (TCKCKEL):** Specifies the amount of time that DRAM clocks need to toggle after CKE goes low (in DRAM Clocks). • For WIO2/LPDDR3, this covers tCPDED. • For LPDDR4, this covers both tCKELCK and tCKELCS. • For DDR3L/DDR4, this is tCKSRE. Note: D-Unit hardware enforces minimum of one SPID clocks after CKEL, any value in this register is the additional time.
18:15	0h RW	**Reserved (RSVD18_15):** Reserved
14:8	0h RW	**Mode Register Read to Any Command Delay (TPSTMRRBLK):** Specifies the quiet time after issuing MRR command (in DRAM Clocks). This is the channel block time for the MR19 command issued as part of LPDDR4 DQS Retraining flow. For all other MR commands this is the rank block time. Note: D-Unit treats MRR as a read and always applies relevant turnaround times, any value programmed in this CR must be greater than those turnaround times for D-Unit to enforce any additional time from MRR to the next read/write.
7	0h RO	**Reserved (RSVD7):** Reserved
6:0	0h RW	**Any Command to Mode Register Read/Write Delay (TPREMRBLK):** Specifies the channel quiet time before issuing MRR/MRW command. (in DRAM clocks). • LPDDR3 Equation: = tXP + tMRRI - 4. • LPDDR4 Equation: = tXP + tMRRI - 8. • DDR3L Equation: = CWL + BL/2 + 1 - 8 • DDR4 Equation: = CWL + BL/2 + 1 - 16. Note: D-Unit treats MRR as a read and MRW/MRS as a write and always applies relevant turnaround times, any value programmed in this CR must be greater than those turnaround times for D-Unit to enforce any additional time from previous read/writes.

3.174 DRAM Timing Register 7A (D_CR_DTR7A) — Offset 1A24h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A24h	D060C06 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	3h RW	**All Bank Precharge to Activate Command Delay [tRPab] (TRPAB):** Specifies the delay between a DRAM Precharge All Bank command and a DRAM Activate command (in DRAM Clocks). Note: This CR should be constrained to a minimum of 4 in LP3 and minimum of 8 in LP4. Note: Derating adds 1.875ns to this timing. • For LPDDR, tRPpb = tRP, tRPab = tRP + 3ns. • For DDR3L, DDR4 and WIO2 8ch tRPpb = tRPab = tRP.
25:23	2h RW	**Mode Register Write to any Command Delay [tMRD/tMRW] (TPSTMRWBLK):** Specifies the quiet time after issuing MRW command (in 8 x DRAM clocks). Note: This time covers for both tMRD and tMRW.
22:16	6h RW	**Write Command to Power Down Delay [tWRPDEN] (TWRPDEN):** Specifies the minimum time between a write command to PowerDown command (in DRAM clocks). Must be at least equal to tWR + tCCD + tWL + 2.
15:9	6h RW	**Read Command to Power Down Delay [tRDPDEN] (TRDPDEN):** Specifies the minimum time between a read command to PowerDown command (in DRAM clocks). Must be at least equal to CL/RL + tDQSCKmax + tCCD + 1.
8:7	0h RO	**Reserved (RSVD8_7):** Reserved
6:0	6h RW	**Row Activation Period [tRAS] (TRAS):** Specifies the minimum delay between the DRAM Activate and Precharge commands to the same bank (in DRAM clocks). Note: Derating adds 1.875ns to this timing.

3.175 DRAM Timing Register 8A (D_CR_DTR8A) — Offset 1A28h

Specifies DRAM timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A28h	CC50A18 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	3h RW	**Minimum Self-Refresh Time [tSR/tCKESR] (TCKESR):** Specifies the minimum time that DRAM should remain in SR (in DRAM clocks).
25:21	6h RW	**Minimum Low Power Mode Residency (LPMDRES):** Specifies the minimum time that PHY should remain in LPMode (in DRAM clocks).
20:15	Ah RW	**Low Power Mode Exit to Clock Enable Delay (LPMDTOCKEDLY):** Specifies the minimum time between the LP Mode exit to the CK stop/tristate deassertion and powerdown exit (in DRAM clocks). Note: Must be equal to t_idle_latency published in the DDRIO PHY and less than 0x3C.
14:8	Ah RW	**Clock Stop to Low Power Mode Delay (CKETOLPMDDLY):** Specifies the time between CK stop/tristate to the Low Power Mode entry. This timing parameter is used to delay Low Power Mode entry (in DRAM clocks). Note: Must be at least equal to t_idle_length parameter published in the DDRIO PHY and less than 0x7C.
7:0	18h RW	**Power Down Idle Timer (PWDDLY):** This is a non-JEDEC timing parameter used to delay powerdown entry (in DRAM clocks).

3.176 D-Unit ODT Control Register A (D_CR_DOCRA) — Offset 1A2Ch

Specifies the parameters to control DRAM ODT.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29	0h RW	**Rank 1 Read ODT Control (R1RDODTCTL):** Specifies the behavior of ODT signals when a Read command is issued to Rank 1. 0 - Read ODT is disabled for Rank 1 1 - Assert ODT to for Rank 0 (non-targeted Rank) Note: This register should be set to 0 for LPDDR3 devices
28	0h RW	**Rank 0 Read ODT Control (R0RDODTCTL):** Specifies the behavior of ODT signals when a Read command is issued to Rank 0. 0 - Read ODT is disabled for Rank 0 1 - Assert ODT to for Rank 1 (non-targeted Rank) Note: This register is reserved for LPDDR3 devices
27:26	0h RW	**Rank 1 Write ODT Control (R1WRODTCTL):** Specifies the behavior of ODT signals when a Write command is issued to Rank 1. 00 - Write ODT is disabled 01 - Assert ODT to Rank 0 (non-targeted Rank) 10 - Assert ODT to Rank 1 (targeted Rank) 11 - Assert ODT to Rank 0 and Rank 1 Note: 10 and 11 are reserved values for LPDDR3
25:24	0h RW	**Rank 0 Write ODT Control (R0WRODTCTL):** Specifies the behavior of ODT signals when a Write command is issued to Rank 0. 00 - Write ODT is disabled 01 - Assert ODT to Rank 0 (targeted Rank) 10 - Assert ODT to Rank 1 (non-targeted Rank) 11 - Assert ODT to Rank 0 and Rank 1 Note: 10 and 11 are reserved values for LPDDR3
23:18	0h RO	**Reserved (RSVD23_18):** Reserved
17:14	0h RW	**Read ODT assertion to de-assertion delay (DDR3L/DDR4) (RDODTSTOP):** Specifies Read ODT assertion to ODT de-assert delay (in DRAM clocks). DDR3L Equation: RDODTSTOP = 6. DDR4 Equation: RDODTSTOP = 5 + tRPRE. Note: Add 1 if DOCRx.RDODTSTART = CL - CWL in 2N mode.
13	0h RO	**Reserved (RSVD13):** Reserved
12:9	0h RW	**Read command to ODT assertion delay (DDR3L/DDR4) (RDODTSTART):** Specifies Read ODT assertion delay after Read Command (in DRAM clocks). DDR3L/DDR4 Equation: RDODTSTART = CL - CWL + tWPRE - tRPRE. Note: In DDR3L/DDR4 2N mode add 1 to enable termination at the start of read data burst.
8:5	0h RW	**Write ODT Assertion to De-assertion Delay (WRODTSTOP):** Specifies number of clocks after ODT assertion that D-Unit deasserts ODT signal (in DRAM clocks). LPDDR3 Equation: WRODTSTOP = RU(tODTon(max)/tCK) + RU((tDQSSmax+tWPST)/tCK) + BL/2 - RD(tODToffmin/tCK) DDR3L Equation: WRODTSTOP = 6. DDR4 Equation: WRODTSTOP = 5 + tWPRE. Note: Add 1 if DOCRx.WRODTSTART = 0 in 2N mode.
4	0h RO	**Reserved (RSVD4):** Reserved
3:0	0h RW	**Write command to ODT assertion delay (WRODTSTART):** Specifies number of clocks after Write command that D-Unit asserts ODT signal (in DRAM clocks). LPDDR3 Equation: WRODTSTART = WL - RU(tODTon(max)/tCK) DDR3L/DDR4 Equation: WRODTSTART = 0 Note: In DDR3L/DDR4 2N mode the value can be set to 0 to assert ODT one DRAM clock earlier than the Write Command (WR) or set to 1 to assert at the same clock as command (CS assertion).

3.177 D-Unit Power Management Control 0 (D_CR_DPMC0) — Offset 1A30h

Specifies the parameters to control D-Unit power management features.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A30h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**Reserved (RSVD31_29):** Reserved
28:24	0h RW	**SUSPEND/SUSPENDP Power Management Message Opcode (SUSPMOP):** DDRIO PHY Power Mode Opcode: After the D-Unit has placed the DRAM devices in Self Refresh/PASR mode as the result of a SUSPEND/SUSPENDP message, it sends this 5-bit value to the DDRIO PHY to tell it which power saving mode it should enter. Changing this register value while in SUSPEND will have no effect. Note: This opcode cannot be a PM state where it disables PHY PLLs i.e PM7 in LPDDR PHY.
23	0h RO	**Reserved (RSVD23):** Reserved
22	0h RW	**PM Message Wait for Clock Gate Enable (SRPMCLKW):** Specifies when it is safe to send PM message to the PHY. When enabled, D-Unit waits for SPID Clock to deassert before sending a PM message on SR entry. • 0: D-Unit will not wait for SPID_clk to deassert before sending the PM message to PHY. • 1: D-Unit will wait for SPID_clk to deassert before sending PM message to the PHY. Note: The value must be 1 when DYNPMOP = 7h.
21:17	0h RW	**Dynamic Self-Refresh Power Management Message Opcode (DYNPMOP):** DDRIO PHY Power Mode Opcode: After the D-Unit has placed the DRAM devices in Self Refresh mode as the result of a Dynamic Self-Refresh, it sends this 5bit value to the DDRIO PHY to tell it which power saving mode it should enter. Changing this register value while in self-refresh will only change the PM state for the next entry in DynSR.
16	0h RW	**Dynamic Self-Refresh Enable (DYNSREN):** When set to 1, the D-Unit will automatically control DRAM Self Refresh entry and exit based on interface state and requests in pending queues. When there is no pending request in the queues and PMI is idle, then the D-Unit will place the DRAM devices in Self Refresh mode. The DRAM devices will be brought out of Self-Refresh when idle conditions don't hold.
15:0	0h RW	**Self-Refresh Entry Delay (SREDLY):** Specifies the minimum time the D-Unit will wait before it enters Dynamic Self-Refresh mode when idle (in 16x DRAM Clocks). Note: The value in this field needs to be minimum of 4 in functional mode and minimum of 50 in PSMI mode.

3.178 D-Unit Power Management Control 1 (D_CR_DPMC1) — Offset 1A34h

Specifies the parameters to control D-Unit power management features.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A34h	10000028 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29	0h RW	**D-Unit Repeaters Clock Gate Disable (RPTCLKGTDIS):** Setting this bit to 0 allows majority of the repeaters between D-Unit and PHY to clock gate when there is no activity in order to save power. • 0 - Enable Repeaters clock gating. • 1 - Disable Repeaters clock gating. Note: This is a de-feature bit and should be set to 0 for normal operation. Note: The value should only change after DRAM Timing Registers (DTR) are programmed.
28	1h RW	**IOSF-SB End Point Clock Gate Disable (SBEPCLKGTDIS):** Setting this bit to 0 enables the clock gating of IOSF-SB End Points in D-Unit and CPGC when there is no IOSF-SB activity in order to save power. • 0 - Enable IOSF-SB EP clock gating. • 1 - Disable IOSF-SB clock gating. Note: This is a de-feature bit and should be set to 0 for normal operation. Note: The value should only change after DRAM Timing Registers (DTR) are programmed.
27	0h RW	**Local Clock Gate Disable (CLKGTDIS):** Setting this bit to 0 allows the majority of the D-Unit clocks to be gated off when there is no activity in order to save power. When set to 1, D-Unit clockgating is disabled. • 0: Enable. • 1: Disable. Note: This is a de-feature bit and should be set to 0 for normal operation. Note: The value should only change after DRAM Timing Registers (DTR) are programmed.
26	0h RW	**Chip Select Tristate Enable (CSTRIST):** • 0: The DRAM CS pins associated with the enabled ranks are never tristated. • 1: The DRAM CS pins are tristated when DRAM clock is stopped or tristated. Note: CS is not tristated when global tristate flow is disabled (DCBR.TRISTDIS = 1).
25:24	0h RW	**Command/Address Tristate (CMDTRIST):** • 00: The DRAM CA pins are never tristated. • 01: The DRAM CA pins are only tristated when all enabled CKE pins are low. • 10: The DRAM CA pins are tristated when not driving a valid command. • 11: Reserved
23:16	0h RW	**Partial Array Self-Refresh Segment Mask (PASR):** This is the Segment Mask used for the MRW to enable PASR during SUSPENDP (Partial Array Self Refresh entry).
15:8	0h RW	**Page Close Timeout Period (PCLSTO):** Specifies the time from the last access of a DRAM page until that page is scheduled to close by sending a Precharge command to DRAM (in 16 x DRAM clocks).
7	0h RW	**Page Close Timeout Disable (PCLSTODIS):** When disabled, D-Unit will not close the DRAM page when idle. • 0: Enable page close timer. • 1: Disable page close timer (Used during DRAM init and DDRIO training).
6	0h RO	**Reserved (RSVD6):** Reserved
5	1h RW	**ODT Tristate Enable (ODTTRIST):** • 0: The DRAM ODT pins associated with the enabled ranks are never tristated. • 1: DRAMs ODT pins are tristated when DRAM clock is stopped or tristated. Note: ODT is not tristated when global tristate flow is disabled (DCBR.TRISTDIS = 1)

Bit Range	Default & Access	Field Name (ID): Description
4:3	1h RW	**Clock Stop/Tristate Enable (ENCKSTP):** Enable/Disable CK Stop/Tristate During Power down. • 00: Disable CK Stop/Tristate During Power down. • 01: Enable CK Stop During Power down. • 10: Enable CK Tristate During Power down. • 11: Reserved Note: CK is not stopped or tristated when global tristate flow is disabled (DCBR.TRISTDIS = 1).
2:1	0h RW	**Low Power Mode Opcode (LPMODEOP):** D-Unit will send the value in this register after it has entered Powerdown Mode and has stopped/tristated the clock. 00: Disable LPMode. Note: LPMODE entry is not possible when global tristate flow is disabled (DCBR.TRISTDIS = 1).
0	0h RW	**Disable Power Down (DISPWRDN):** Setting this bit to 1 disables dynamic control of DRAM Power-Down entry and exit by keeping the CKE pins driven high. BIOS may set it to 1 during DRAM initialization and DDRIO training. This bit should be set to 0 for normal operation. • 0: The D-Unit dynamically controls the CKE pins to place the DRAM devices in Power Down mode and bring them out of Power Down mode. • 1: The D-Unit constantly drives the CKE pins high to keep the DRAM devices from entering Power Down mode when ranks are idle. Note: This bit is overridden if CKEMODE = 1. This bit does not control CKE behavior on SR entry/exit.

3.179 DRAM Refresh Control (D_CR_DRFC) — Offset 1A38h

Specifies the parameters to control scheduling of refresh commands.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A38h	20001750 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	1h RW	**Maintenance Operation Channel Block Time (MNTCHNBLKTIME):** Specifies the amount of time that D-unit will block the channel before any DRAM maintenance operation is issued. (in SPIDclk)
28:25	0h RW	**Maintenance Operation Delay CAS Count (MNTDLYCASCOUNT):** When a critical read request is pending in RPQ and a maintenance operation (MRR, ZQCal, Ref, etc.) needs to be performed, D-unit delays the maintenance operation and allows this many read or write requests to be scheduled before allowing the maintenance operation. Note: This mode does not apply to Panic refreshes.
24:22	0h RO	**Reserved (RSVD24_22):** Reserved
21	0h RW	**Disable Refresh Debt Clear (DISREFDBTCLR):** When set, D-Unit will not clear refresh debt before Self Refresh SR Entry: • 0: D-Unit sends all postponed REF commands to DRAM before it enters Self Refresh. • 1: D-Unit enters SR without clearing the Refresh Debt (for Debug only).

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**Refresh Skew Disable (REFSKWDIS):** Disables Skewing of Refresh Counting between Ranks. Each rank has its own refresh counter. By default incrementing these refresh counters are skewed by 1/2 the tREFI period. Setting this bit to a 1 disables this feature and all refresh counters will increment at the same time per tREFI period. Skewing the tREFI counters can improve performance since traffic to all ranks does not have to be blocked to perform refresh. • 0: Incrementing the refresh counters are skewed by 1/2 tREFI period. • 1: All refresh counters will increment at the same time per tREFI period.
19:18	0h RO	**Reserved (RSVD19_18):** Reserved
17:16	0h RO	**Reserved (RSVD17_16):** Reserved
15	0h RW	**Extra Refresh Debit (EXTRAREFDBT):** When set to 1, D-Unit adds one extra refresh debit (for a total of two) on Self-refresh exit.
14:12	1h RW	**Minimum Refresh Rate (MINREFRATE):** Ensures that refresh rate never drops below a certain limit regardless of TQ polling. • 000: Stop issuing refresh commands and accumulating refresh debits. (does not stop tREFI counter). • 001: 0.25x refresh rate (i.e. 4x tREFI same as no limit). • 010: 0.5x refresh rate (i.e. 2x tREFI). • 011: 1x refresh rate (i.e. 1x tREFI). • 100: 2x refresh rate (i.e. 0.5x tREFI). • 101: 4x refresh rate (i.e. 0.25x tREFI). • 110: 4x refresh rate with derating forced on i.e. 0.25x tREFI. • 111: Reserved.
11:8	7h RW	**Refresh Panic Watermark (REFWMPNC):** When the Refresh counter per rank is greater than this value, the D-Unit will send a REF command to the rank regardless of pending requests. Note: REFWMPNC must be greater than or equal to REFWMHI and greater than 2, Max Value must be less than 8 to not violate 9xtREFI JEDEC requirement.
7:4	5h RW	**Refresh High Watermark (REFWMHI):** When the Refresh counter per rank is greater than this value, the D-Unit will send a REF command to the rank if there is no critical priority requests in the pending queues. Note: Value must be greater or equal to 1 and less than or equal to REFWMPNC.
3:1	0h RO	**Reserved (RSVD3_1):** Reserved
0	0h RW	**Opportunistic Refresh Disable (OREFDIS):** Disable opportunistic scheduling of refresh. • 0: D-Unit will send a REF command only if there is no pending request to that rank. • 1: D-Unit will not send any opportunistic refreshes. Refresh commands are only sent when the refresh counter is greater than REFWMHI. Note: When set, DISREFDBTCLR must also be set to be able to enter SR.

3.180 D-Unit Scheduler Control (D_CR_DSCH) — Offset 1A3Ch

Specifies parameters to control scheduling of commands to DRAM.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A3Ch	3901C08 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Early Write DQ Enable (EARLY_DQ_EN):** Enables D-unit to send the write data to the PHY one clock earlier than WL value (WL - 1). Note: Applicable to DDR3L and DDR4 only.
30:29	0h RW	**BGF Early Read Data Valid (BGF_EARLY_RDDATA_VALID):** Specifies the number of clocks the D-Unit sends the read data valid through the BGF earlier as compared to the data. • 00: Always write read valid in same SPID clock as data (STATIC_0). • 01: Always write read valid one SPID clock before data (STATIC_1). • 10: Write read valid up to 2 SPID clocks before data (DYNAMIC). • 11: Reserved
28:27	0h RW	**SPID Early Read Data Valid (SPID_EARLY_RDDATA_VALID):** Specifies the delay in SPID clocks from RDDATA_VALID assertion to actual data on SPID. The value should match what is programmed in DDRIO (PHY).
26:21	1Ch RW	**Write Pending Queue Count (WPQCOUNT):** Used to limit the number of available slots in Write Pending Queue/ Write Data Buffer. WPQCOUNT will only recognize changes when PMI ISM is not active.
20:16	10h RW	**Read Pending Queue Count (RPQCOUNT):** Used to limit the number of entries in Read Pending Queue. RPQCOUNT will only recognize changes when PMI ISM is not active.
15	0h RO	**Reserved (RSVD15):** Reserved
14:10	7h RW	**Read Return Data Additional Credits (BLKRDBF_ADD_RDDATA_CR):** Number of additional full cacheline (64B) read data return credits exposed to D-Unit when BLKRDBF is set. Note: The value in this field has no effect on Read return credits when BLKRDBF is not set.
9:8	0h RW	**In-Order Mode (INORDERMODE):** • 0h: In order mode disabled: Commands are sent out of order. • 1h: Partial in order mode: Read and Write CAS commands are sent in the order they were received. ACT and PRE can go out of order. • 2h: Full in order mode serialized test: All DRAM commands CAS ACT PRE associated with a PMI request are issued to DDR before any DRAM commands for a subsequent PMI request. • 3h: Reserved. In order modes should be enabled during init/training/CPGC testing. Should never be changed while the D-Unit queues are nonempty. For WIO2, when in-order mode is enabled (01 or 10), D_CR_DSCH_BYPASSEN must be set to 0
7	0h RW	**Idle Bypass Mode Enable (BYPASSEN):** When set new page hit/empty read requests will bypass D-Unit pipeline stages to save latency 0 - Disable Idle Bypass 1 - Enable Idle Bypass Note: Only applies to WIO2, this bit is reserved for other technologies
6	0h RW	**Block When RDB Full (BLKRDBF):** When set D-Unit stops scheduling new read commands to DRAM when the read data buffer (RDB) is full.
5:4	0h RW	**Stretch Mode (STRETCHMODE):** When stretch mode is enabled, commands are initiated only on Phase 0 of SPIDClk. • 00: Stretch mode is disabled. • 01: Commands are initiated on Phase 0 of every SPID clocks. • 10: Commands are initiated on Phase 0 of even SPID clocks. • 11: Commands are initiated on Phase 0 of odd SPID clocks.
3:0	8h RW	**Masked Write Turnaround Delta (TMWR_TA_DELTA):** The value in this register is subtracted from Masked Write to Read, Masked Write to Write and Masked Write to Masked Write turnaround times to account for half BL MWr commands in LPDDR4 and WIO2. • LPDDR4: = MWr tCCD = MWr BL/2 = 8. • WIO2: = NA.

3.181 DRAM Calibration Control (D_CR_DCAL) — Offset 1A40h

Specifies parameters to control ZQ Calibration.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A40h	1057 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**ZQ Calibration Type (ZQCALTYPE):** Determines whether the ZQ Calibration is a long or short calibration command (due to ZQCALSTRT). 0: Short calibration (ZQCS). 1: Long calibration (ZQCL).
30	0h RW/V	**ZQ Calibration Start Rank 1 (ZQCALSTRTR1):** Set this bit to 1 to start the ZQ calibration sequence on Rank 1. This bit will remain a 1 until the ZQ calibration is complete for rank 1, then it will return to 0. 0: ZQ calibration is done. 1: ZQ calibration has started and is in progress.
29	0h RW/V	**ZQ Calibration Start Rank 0 (ZQCALSTRTR0):** Set this bit to 1 to start the ZQ calibration sequence on Rank 0. This bit will remain a 1 until the ZQ calibration is complete for rank 0, then it will return to 0. 0: ZQ calibration is done. 1: ZQ calibration has started and is in progress.
28:23	0h RO	**Reserved (RSVD28_23):** Reserved
22:21	0h RW	**Self-Refresh Exit ZQ Calibration Control (SRXZQC):** • 00: On DynSR exit ZQ timer determines the ZQ type. When the state is lost (i.e due to AutoPG/S0ix) ZQCL is always performed. • 01: Always perform ZQCL after self refresh exit. In LPDDR4, ZQ with traffic blocked. • 10: Always perform ZQCS on SR exit. For LPDDR4, ZQ while traffic is allowed. • 11: No ZQCL commands are sent (it disables ZQCAL commands on SR exit).
20:18	0h RO	**Reserved (RSVD20_18):** Reserved
17	0h RW	**ZQ Calibration Mode (ZQCLMODE):** Specifies how ZQCal commands are sent to different ranks. • 0: ZQCal commands are sent in parallel to all ranks. • 1: ZQCal commands are sent serially to each rank.
16	0h RW	**Periodic ZQ Calibration Disable (ZQCDIS):** • 0: Periodic ZQ Calibration is Enabled. • 1: Disable periodic ZQ Calibration.
15:14	0h RO	**Reserved (RSVD15_14):** Reserved
13:0	1057h RW	**ZQ Calibration Interval (ZQINT):** Specifies the time interval between two ZQCS (LPDDR3/DDR3L/DDR4) or ZQ Start (LPDDR4) commands to a DRAM device. (in RTC 32.8KHz clocks)

3.182 DRAM Mode Registers Shadow Copy (D_CR_MR_SHADOW) — Offset 1A48h

This register contains a copy of Mode Regsiters in the DRAM so that D-unit can only modify certain bits without affecting other values.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A48h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:16	0h RW	**MR Value 2 (MR_VALUE2):** MR3 Shadow Register (DDR4): BIOS writes the correct value of MR3 register in DDR4 into this field at boot time. D-Unit modifies one bit and rewrites the MR3 into DDR4 to enter and exit MPR mode.
15:14	0h RO	**Reserved (RSVD15_14):** Reserved
13:0	0h RW	**MR Value (MR_VALUE):** MR3 Shadow Register (WIO2): BIOS sets the value of this field at boot time based on the DRAM device configuration. D-Unit merges the value in MR3_THERM_OFFSET with this field and writes the result into DRAM MR3 MR2 Shadow Register (DDR3L): BIOS writes the correct value of MR2 register in DDR3L into this field at boot time. D-Unit modifies one bit and rewrites the MR2 into DDR3L DRAM before SR entry. MR4 Shadow Register (DDR4): BIOS writes the correct value of MR4 register in DDR4 into this field at boot time. D-Unit modifies one bit and rewrites the MR4 into DDR4 DRAM after temperature read out.

3.183 VNN Scaling Timer Control (D_CR_VNNTIMER) — Offset 1A4Ch

Specifies parameters for VNN Scaling Timer in D-Unit. The values in this register will be set by P-code during VNN scaling period.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A4Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**VNN Scaling Timer Enable (VNN_TIMER_EN):** • 0: The D-Unit VNN Scaling Timer is disabled. • 1: The D-Unit VNN Scaling Timer is enabled.
30:12	0h RO	**Reserved (RSVD30_12):** Reserved
11:0	0h RW	**VNN Timer Time (VNN_TIMER_TIME):** The final timer value (in 16 x DRAM clocks).

3.184 Periodic DRAM Temperature Polling Control (TQ) (D_CR_TQCTL) — Offset 1A50h

Specifies the control for periodic temperature monitoring and control of DRAM device.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A50h	6C000008 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	3h RW/V	**TQ Data Rank 1 (TQDATAR1):** If Rank 1 is disabled, this value will remain zero. This field contains the data of the last temperature sensor read from Rank 1 DRAM Mode Register. It is overwritten with each command.
28:26	3h RW/V	**TQ Data Rank 0 (TQDATAR0):** This field contains the data of the last temperature sensor read from Rank 0 DRAM Mode Register. It is overwritten with each command.
25:22	0h RO	**Reserved (RSVD25_22):** Reserved
21:8	0h RW	**TQ Poll Period (TQPOLLPER):** This sets the frequency by which the D-Unit initiates temperature sensor read from DRAM mode register to determine required refresh rate (in 4x tREFI units).
7	0h RW	**Temperature Controlled Refresh Range Enable (DDR4 Only) (TCRREN):** When set, after a DRAM temperature read out (MPR sequence), D-unit writes a 1 to bit 2 of MR_SHADOW.MR_VALUE when temperature sensor read out for that rank indicates a value higher than 0x1, and writes a 0 to that bit otherwise. The new MR_VALUE is then written into MR4 of DDR4 for each enabled rank.
6	0h RW/V	**TQ Poll Start Rank 1 (TQPOLL_START_R1):** When set Dunit will initiate temperature sensor read for Rank 1. Hardware will clear the bit once the MR read command is issued for this rank. Note: TQPOLLEN mus be 0 before this bit is set.
5	0h RW/V	**TQ Poll Start Rank 0 (TQPOLL_START_R0):** When set Dunit will initiate temperature sensor read for Rank 0. Hardware will clear the bit once the MR read command is issued for this rank. Note: TQPOLLEN mus be 0 before this bit is set.
4	0h RW	**Self Refresh Temperature Range Enable (DDR3 Only) (SRTEN):** When set, before every Self refresh entry, D-Unit writes a 1 to bit 7 of MR_SHADOW.MR_VALUE when TQDATA for that rank indicates a value higher then 0x3, and writes a 0 to that bit otherwise. The new MR_VALUE is then written into MR2 of DDR3 for each enabled rank.
3	1h RW	**Enable Dynamic Timing Derating (ENDERATE):** When set to 1, the Dynamic Timing Derating is enabled. When the D-Unit determines (via TQ polling) that the DRAM requires timing derating in addition to refresh interval adjustment, the D-Unit will automatically adjust the relevant timing parameters.
2	0h RW	**Enable TQ Data Push (TQDATAPUSHEN):** When set to 1, D-Unit pushes the data form the last temperature sensor read to a punit register.
1	0h RW	**Enable TQ Poll on Self-Refresh Exit (TQPOLLSREN):** This bit enables temperature sensor read on Self Refresh Exit. If disabled, D-Unit will not initiate MR Read sequence to read temperature value on Self-Refresh exit.
0	0h RW	**Enable Periodic TQ Poll (TQPOLLEN):** This bit enables periodic TQ Poll. If disabled, D-Unit will not read the DRAM's temperature sensor value periodically.

3.185 Temperature Offset Control (D_CR_TQOFFSET) — Offset 1A54h

Specifies temperature offset and refresh rate adjustments requested by software.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A54h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD31_16):** Reserved
15:11	0h RO	**Reserved (RSVD15_11):** Reserved
10:8	0h RW	**MR4 Adder (MR4_ADDER):** D-Unit adds the value of this field to TQDATA read from MR4 the resulting value is used to control refresh rate and AC timing derating.
7:3	0h RO	**Reserved (RSVD7_3):** Reserved
2	0h RW/V	**MR3 Offset Update (MR3_OFFSET_UPDATE):** When set, D-Unit writes the merged value of MR3_VALUE and MR3_THERM_OFFSET into MR3 of DRAM. D-Unit clears this bit once the value is written.
1:0	0h RW	**MR3 Thermal Offset (MR3_THERM_OFFSET):** (WIO2 only) Everytime MR3_OFFSET_UPDATE is set, D-Unit merges the value in this field with MR3_VALUE and writes the result to MR3 in DRAM

3.186 D-Unit Control Operations (D_CR_DCO) — Offset 1A58h

Specifies D-Unit initialization and control operation.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1A58h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/V	**Initialization Complete (IC):** Indicates that initialization of the D-Unit has been completed. Memory accesses are permitted and maintenance operation begins. Until this bit is set to a 1, the memory controller will not accept DRAM requests from the Bunit/GSA/2LM (PMI ISMs will not leave idle). Note: Set this bit to 1 only when all other D-Unit registers have been configured. Usually set at the last configuration step by BIOS on cold/warm reset. D-Unit hardware sets this bit on SR exit.
30	0h RO/V	**DDRIO PHY Initialization Complete (DIOIC):** Status indication that the DDRIO PHY initialization is complete reflects the status spid_init_complete signal.
29	0h RO	**Reserved (RSVD29):** Reserved
28	0h RW	**PMI Control Select (PMICTL):** • 0: D-Unit PMI is connected to Bunit/GSA/2LM. • 1: D-Unit PMI is connected to CPGC. Note: D_CR_DSCH_BYPASSEN must be set to 0 in CPGC mode. Note: PMI must be idle and D-Unit BGF_RUN = 0 before changing the value in this register.
27:8	0h RO	**Reserved (RSVD27_8):** Reserved
7:4	0h RO	**Reserved (RSVD7_4):** Reserved
3	0h RW	**Enable PSMI Mode (PSMIEN):** When enabled, D-Unit will synchronize clock crossing signals. • 0: PSMI Mode is disabled. • 1: PSMI Mode is enabled. Note: Change only allowed when D-Unit is idle.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Maintenance Reset (MNTRST):** Writing a 1 to this field resets all maintenance timers. Clears all states and also clears refresh debt queues. This bit must be cleared by software after at least 3 SPID clocks.
1	0h RW	**Enable Maintenance Operations (MNTEN):** Setting this field to 1 enables all maintenance operations. When DCO.IC is set, the maintenance operations are enabled irrespective of the value of this field.
0	0h RO	**Reserved (RSVD0):** Reserved

3.187 Data Scrambler (D_CR_SCRAMCTRL) — Offset 1AA4h

Specifies parameters to control data scrambling in D-Unit.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1AA4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Enable Data Scrambler (SCRM_EN):** When set to 1, data scrambling is enabled. When set to 0, data scrambling is disabled. Should be set before D_CR_BGF_CTL_BGF_RUN is set to 1.
30	0h RO	**Reserved (RSVD30):** Reserved
29:28	0h RW	**Scrambler Clock Gate Select (CLOCKGATE):** This field controls how the scrambler output code is clock gated to reduce power. • 00: Clock gate disabled. • 01: Clock Gate every 2 cycles. • 10: Clock Gate every 3 cycles. • 11: Clock Gate every 4 cycles.
27:16	0h RO	**Reserved (RSVD27_16):** Reserved
15:0	0h RW	**Scrambling Key (KEY):** Sets the key for the scrambler. The key should be a random value that is set following each cold boot.

3.188 Error Injection Address Register (D_CR_ERR_INJ) — Offset 1AACh

Contains the target address for ECC error injection.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1AACh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD31):** Reserved
30:1	0h RW	**Error Injection Target Address (ADDRESS):** Specifies the PMI address of the write transaction to be injected with the error. Only applicable to Write transactions. Read/ under-fill read of the partial write operation is not affected.
0	0h RO	**Reserved (RSVD0):** Reserved

3.189 Error Injection Control Register (D_CR_ERR_INJ_CTL) — Offset 1AB0h

Controls injecting correctable or uncorrectable errors into the write requests specified by target address.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1AB0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RSVD31_4):** Reserved
3	0h RW	**Error Injection Type Higher 32B (SEL_HI):** 0 - Uncorrectable Error (UE) is armed for write address matching to inject UE by using the same poisoning scheme, i.e. inverting corresponding write ECC[6:0] on every QW of the 32B data. 1 - Correctable Error (CE) is armed for write address matching to inject CE by inverting corresponding write ECC[0] on every QW of the 32B data.
2	0h RW	**Error Injection Enable Higher 32B (EN_HI):** When set the error injection is continuously armed for higher 32B of D_CR_ERR_INJ_ADDR write address matching until it is cleared.
1	0h RW	**Error Injection Type Lower 32B (SEL_LO):** 0 - Uncorrectable Error (UE) is armed for write address matching to inject UE by using the same poisoning scheme, i.e. inverting corresponding write ECC[6:0] on every QW of the 32B data. 1 - Correctable Error (CE) is armed for write address matching to inject CE by inverting corresponding write ECC[0] on every QW of the 32B data.
0	0h RW	**Error Injection Enable Lower 32B (EN_LO):** When set, the error injection is continuously armed for lower 32B of D_CR_ERR_INJ_ADDR write address matching until it is cleared.

3.190 Error Log Register (D_CR_ERR_ECC_LOG) — Offset 1AB4h

Detected ECC errors are captured in this register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1AB4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/V	**CLEAR:** Setting this bit to one clears all fields in this register, including itself.
30:29	0h RW	**PMI VISA Byte Select (ECC_VISA):** Select ECC or PMI byte on VISA : • 00: ECC byte, • 01: PMI Data Byte [7:0], • 10: PMI Data Byte [63:56], • 11: PMI Data Byte [255:248]
28	0h RW/V	**Correctable Single-bit Error (CERR):** This bit is set when a correctable single-bit error occurs on a memory read data transfer. When this bit is set, the address that caused the error and the error syndrome are also logged and they are locked to further single bit errors, until this bit is cleared. A multiple bit error that occurs after this bit is set will override the address/error syndrome information.
27	0h RW/V	**Uncorrectable Multiple-bit Error (MERR):** This bit is set when an uncorrectable multiple-bit error occurs on a memory read data transfer. When this bit is set, the address that caused the error and the error syndrome are also logged and they are locked until this bit is cleared.
26:25	0h RW/V	**Error Burst Number (ERR_BURST):** Burst number (in BL8) of the error within a chunk.
24	0h RW/V	**Error Chunk Number (ERR_CHUNK):** Chunk number of the error. 0 - lower 32B chunk has error if MERR/CERR is set 1 - higher 32B chunk has the error if MERR/CERR is set
23:16	0h RW/V	**Quad Word ECC Syndrome (SYNDROME_QW):** ECC Syndrome for a QW (64 bit) within 32B Address
15:0	0h RW/V	**Request Tag (TAG):** Read Return Tag matches with the PMI Request Tag which triggered the error log.

3.191 D-Unit Fuse Status (D_CR_DFUSESTAT) — Offset 1ABCh

Contains the values read from D-Unit fuses.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1ABCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD31_16):** Reserved
15:0	0h RO/V	**D-Unit Fuse Status (FUSESTAT):** D-Unit fuse bits are captured into this register and are available to be read. • [0]: fus_dun_ecc_dis. • [3:1]: fus_dun_max_supported_device_size[2:0]. • [4:4]: fus_dun_lpddr3_dis. • [5:5]: fus_dun_lpddr4_dis. • [6:6]: fus_dun_wio2_dis. • [7:7]: fus_dun_ddr3l_dis. • [8:8]: fus_dun_ddr4_dis. • [15:9]: reserved.

3.192 Major Mode Control (D_CR_MMC) — Offset 1B24h

Specifies parameters to control read/write major mode operation and transitions.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1B24h	2B01A518 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:27	5h RW	**RAW Conflict Read Priority for WMM Transition (RAW_WMM):** If a conflict read reaches this priority (or greater depending on access class occupancy), WMM will be triggered to unblock the corresponding write. D-Unit will stay in WMM until corresponding write is issued. Note: The value in this bit must not be higher than lowest terminal priority level of each access class.
26	0h RO	**Reserved (RSVD26):** Reserved
25:23	6h RW	**Read Isoch Trigger Priority (RIMPRIO):** If any read in the RPQ is at this programmable priority, RIM is triggered.
22:18	0h RO	**Reserved (RSVD22_18):** Reserved
17:12	1Ah RW	**Write Isoch Threshold (WIMTHRS):** When the number of entries in WPQ is greater than or equal to this value (higher than WMM entry watermark, less than WPQ size), it triggers write isoch mode (WIM).
11:6	14h RW	**Write Major Mode Exit Watermark (WMMEXIT):** When the number of entries in WPQ is less than this value, the D-Unit will switch back to read major mode.
5:0	18h RW	**Write Major Mode Entry Watermark (WMMENTRY):** When the number of entries in WPQ is greater than or equal to this value, the D-Unit will switch to write major mode (WMM). Note: the value must not be set to 0.

3.193 Major Mode RD/WR Counter (Set A and B) (D_CR_MMRDWR_AB) — Offset 1B28h

Minimum read and maximum write counter control. This register defines the minimum number of reads in RMM and maximum number of writes in WMM before a mode transition happens.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1B28h	1020408 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (RSVD31_26):** Reserved
25:20	10h RW	**Max Writes B (MAXWRB):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set B).

Bit Range	Default & Access	Field Name (ID): Description
19:14	8h RW	**Min Reads B (MINRDB):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set B).
13:12	0h RO	**Reserved (RSVD13_12):** Reserved
11:6	10h RW	**Max Writes A (MAXWRA):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set A).
5:0	8h RW	**Min Reads A (MINRDA):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set A).

3.194 Major Mode RD/WR Counter (Set C and D) (D_CR_MMRDWR_CD) — Offset 1B2Ch

Minimum read and maximum write counter control. This register defines the minumum number of reads in RMM and maximum number of writes in WMM before a mode transition happens (sets C and D).

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1B2Ch	1020408 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (RSVD31_26):** Reserved
25:20	10h RW	**Max Writes D (MAXWRD):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set D).
19:14	8h RW	**Min Reads D (MINRDD):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set D).
13:12	0h RO	**Reserved (RSVD13_12):** Reserved
11:6	10h RW	**Max Writes C (MAXWRC):** Maximum number of writes D-Unit can send in WMM mode before returning to RMM (set C).
5:0	8h RW	**Min Reads C (MINRDC):** Minimum number of reads that has to be serviced before a switch to WMM is allowed (set C).

3.195 Access Class Initial Priority (D_CR_ACCIP) — Offset 1B30h

Each field of this register defines the initial priority of one access class.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1B30h	17C2 h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD31_15):** Reserved
14:12	1h RW	**Access Class 4 Initial Priority (AC4IP):** Initial priority level of read requests coming with access class 4.
11:9	3h RW	**Access Class 3 Initial Priority (AC3IP):** Initial priority level of read requests coming with access class 3.
8:6	7h RW	**Access Class 2 Initial Priority (AC2IP):** Initial priority level of read requests coming with access class 2.
5:3	0h RW	**Access Class 1 Initial Priority (AC1IP):** Initial priority level of read requests coming with access class 1.
2:0	2h RW	**Access Class 0 Initial Priority (AC0IP):** Initial priority level of read requests coming with access class 0.

3.196 Access Class 0 Priority Promotion Control (D_CR_RD_PROM0) — Offset 1B34h

This register defines the priority promotion policy for access class 0. Each field of this register defines the number of Cases that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1B34h	1F52940 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
24:20	1Fh RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	Ah RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	Ah RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	Ah RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.197 Access Class 1 Priority Promotion Control (D_CR_RD_PROM1) — Offset 1B38h

This register defines the priority promotion policy for access class 1. Each field of this register defines the number of Cases that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the associated level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1B38h	14000000 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	Ah RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
24:20	0h RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	0h RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	0h RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	0h RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.198 Access Class 2 Priority Promotion Control (D_CR_RD_PROM2) — Offset 1B3Ch

This register defines the priority promotion policy for access class 2. Each field of this register defines the number of Cases that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1B3Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

Bit Range	Default & Access	Field Name (ID): Description
24:20	0h RW	**Priority 5 Residency (P5RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
19:15	0h RW	**Priority 4 Residency (P4RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
14:10	0h RW	**Priority 3 Residency (P3RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
9:5	0h RW	**Priority 2 Residency (P2RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of CASes that pass before a request in this priority promotes to the next priority level.

3.199 Access Class 3 Priority Promotion Control (D_CR_RD_PROM3) — Offset 1B40h

This register defines the aging policy for access class 3. Each field of this register defines the number of Cases that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1B40h	1F29400 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of Cases that pass before a request in this priority promotes to the next priority level.
24:20	1Fh RW	**Priority 5 Residency (P5RES):** Number of Cases that pass before a request in this priority promotes to the next priority level.
19:15	5h RW	**Priority 4 Residency (P4RES):** Number of Cases that pass before a request in this priority promotes to the next priority level.
14:10	5h RW	**Priority 3 Residency (P3RES):** Number of Cases that pass before a request in this priority promotes to the next priority level.
9:5	0h RW	**Priority 2 Residency (P2RES):** Number of Cases that pass before a request in this priority promotes to the next priority level.
4:0	0h RW	**Priority 1 Residency (P1RES):** Number of Cases that pass before a request in this priority promotes to the next priority level.

3.200 Access Class 4 Priority Promotion Control (D_CR_RD_PROM4) — Offset 1B44h

This register defines the aging policy for access class 3. Each field of this register defines the number of Cases that pass before the request is promoted to the next priority level. A value of 31 indicates the request should never be promoted to the next level and the request has reached its maximum priority.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1B44h	1F5294A h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD31_30):** Reserved
29:25	0h RW	**Priority 6 Residency (P6RES):** Number of Cases that pass before a request in this priority promotes to the next priority level.
24:20	1Fh RW	**Priority 5 Residency (P5RES):** Number of Cases that pass before a request in this priority promotes to the next priority level.
19:15	Ah RW	**Priority 4 Residency (P4RES):** Number of Cases that pass before a request in this priority promotes to the next priority level.
14:10	Ah RW	**Priority 3 Residency (P3RES):** Number of Cases that pass before a request in this priority promotes to the next priority level.
9:5	Ah RW	**Priority 2 Residency (P2RES):** Number of Cases that pass before a request in this priority promotes to the next priority level.
4:0	Ah RW	**Priority 1 Residency (P1RES):** Number of Cases that pass before a request in this priority promotes to the next priority level.

3.201 Deadline Threshold (D_CR_DL_THRS) — Offset 1B48h

Specifies when the request with initial priority 0 get promoted to a higher priority level.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1B48h	6 h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD31_11):** Reserved
10:0	6h RW	**Deadline Threshold (DEADLINE_THRS):** A requests with initial priority of 0 will exit priority 0 when its deadline is equal or less than this value plus current time. This field does not affect the priority of any requests in access classes with initial priority bigger than 0.

3.202 Major Mode Blocking Rules Control (D_CR_MM_BLK) — Offset 1B4Ch

This register controls blocking rules enforced in RMM and WMM.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1B4Ch	1800 h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD31_25):** Reserved
24	0h RW	**WMM Regular Rule 1 (WMM_REG_R1):** Disable WMM unsafe write page hits block safe write page misses same bank.
23:20	0h RO	**Reserved (RSVD23_20):** Reserved
19	0h RW	**WMM Priority Rule 4 (WMM_PRIO_R4):** Disable WMM unsafe priority 1 read miss block write hit to same bank. Note: This rule does not block the bank that is being blocked by WMM_PRIO_R3. Priority rules 1, 3 and 4 should be enabled/disabled together.
18	0h RW	**WMM Priority Rule 3 (WMM_PRIO_R3):** Disable WMM unsafe priority 1 write hit block write miss to same bank. Note: This rule does not block the bank that is being blocked by WMM_PRIO_R1. Priority rules 1,3 and 4 should be enabled/disabled together.
17	0h RW	**WMM Priority Rule 2 (WMM_PRIO_R2):** Disable WMM CAS block rule.
16	0h RW	**WMM Priority Rule 1 (WMM_PRIO_R1):** Disable WMM unsafe top priority 1 write miss block write hit same bank. Priority rules 1, 3 and 4 should be enabled/disabled together.
15:14	0h RO	**Reserved (RSVD15_14):** Reserved
13	0h RW	**RMM Regular Rule 6 (RMM_REG_R6):** Disable RMM unsafe write page hits block safe write page misses same bank.
12	1h RW	**RMM Regular Rule 5 (RMM_REG_R5):** Disable RMM unsafe read page miss block all safe and unsafe write page hit to the same bank. Note: This field must not be set to 0 (enabled) if RMM_REG_R4 is also 0.
11	1h RW	**RMM Regular Rule 4 (RMM_REG_R4):** Disable RMM unsafe write page hit block safe read page miss same bank. Note: This field must not be set to 0 (enabled) if RMM_REG_R5 is also 0.
10	0h RW	**RMM Regular Rule 3 (RMM_REG_R3):** Disable RMM unsafe read page hit block safe read and write page miss same bank. Note: This rule does not block the bank that is being blocked by RMM_PRIO_R3 and RMM_PRIO_R1.
9	0h RW	**RMM Regular Rule 2 (RMM_REG_R2):** Disable RMM unsafe read page empty block safe write page empty same rank.
8	0h RW	**RMM Regular Rule 1 (RMM_REG_R1):** Disable RMM unsafe read page hit block safe write page hit same rank.
7:4	0h RO	**Reserved (RSVD7_4):** Reserved
3	0h RW	**RMM Priority Rule 4 (RMM_PRIO_R4):** Disable RMM unsafe critical read miss block read and write hit to same bank. Note: This rule does not block the bank that is being blocked by RMM_PRIO_R3. Priority rules 1, 3 and 4 should be enabled/disabled together.
2	0h RW	**RMM Priority Rule 3 (RMM_PRIO_R3):** Disable RMM unsafe critical read hit block read and write miss to same bank. Note: This rule does not block the bank that is being blocked by RMM_PRIO_R1. Priority rules 1, 3 and 4 should be enabled/disabled together.
1	0h RW	**RMM Priority Rule 2 (RMM_PRIO_R2):** Disable RMM CAS block rule.
0	0h RW	**RMM Priority Rule 1 (RMM_PRIO_R1):** Disable RMM unsafe top critical read miss block read and write hit same bank. Note: Priority rules 1, 3 and 4 should be enabled/disabled together.

3.203 DRAM Self-Refresh Command (D_CR_DRAM_SR_CMD) — Offset 1B54h

Self refresh command register to allow sending WAKE and SUSPEND messages to D-Unit. (Only one bit can be set at a time). Posted writes to this register are not completed until hardware clears the field.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1B54h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RSVD31_4):** Reserved
3	0h RW/V	**SUSPENDP:** A SUSPENDP message will put the DRAM into self-refresh mode. The D-Unit will complete servicing outstanding memory requests and flush all queued Refresh commands to DRAM before putting the DRAM in self refresh mode. Finally, a PM message will be sent to the PHY. The bit is cleared by hardware after the PHY indicates the transition requested in the PM message has been completed. D-Unit will perform an MRW to MR17 with an opcode as defined by DPMC0.PASR before it places the DRAM into Self-Refresh.
2	0h RW/V	**SUSPEND:** A SUSPEND message will put the DRAM into self-refresh mode. The D-Unit will complete servicing outstanding memory requests and flush all queued Refresh commands to DRAM before putting the DRAM in Self Refresh mode. Finally, a PM message will be sent to the PHY. The bit is cleared by hardware only after the PHY indicates the transition requested in the PM message has been completed. Note: When COLDWAKE is set prior of setting this bit the DRAM will not be placed in SR.
1	0h RO	**Reserved (RSVD1):** Reserved
0	0h RW/V	**WAKE:** Take PHY out of PM states and wakes the DRAM out of self refresh mode. The bit is cleared by hardware only when the DRAM has exited out of self refresh mode and is accessible. Note: When COLDWAKE is set prior of setting this bit the D-Unit will not send SR exit command and will not set the DCO.IC bit.

3.204 DQS Retraining Control (D_CR_DQS_RETRAINING_CTL) — Offset 1B80h

LPDDR4 DQS Retraining control register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1B80h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DQS Periodic Retraining Interval (DQS_RETRAIN_INT):** This sets the frequency by which the D-Unit initiates periodic retraining (in 1x NREFI).
15:14	0h RO	**Reserved (RSVD15_14):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:4	0h RW	**DQS Oscillator Runtime (DQS_OSC_RT):** After D-Unit starts DQS oscillator, it must wait this amount of time before being able to read the value in MR18 and MR19 (in 16x DRAM clocks). Value in this register must be at least equal to DRAM's MR23 value. + tOSCO.
3	0h RW/V	**DQS Retrain Start Rank 1 (DQS_RETRAIN_START_R1):** When set, Dunit will initiate LPDDR4 DQS training sequence on Rank1. Hardware will clear bit when sequence completes (after issuing MR19). Note: DQS_RETRAIN_EN must be 0 before setting this bit.
2	0h RW/V	**DQS Retrain Start Rank 0 (DQS_RETRAIN_START_R0):** When set, Dunit will initiate LPDDR4 DQS training sequence on Rank0. Hardware will clear bit when sequence completes (after issuing MR19). Note: DQS_RETRAIN_EN must be 0 before setting this bit.
1	0h RW	**DQS Retrain SRX Exit (DQS_RETRAIN_SRX_EN):** Enable retraining on SR exit. This bit enables LPDDR4 DQS retraining on Self Refresh Exit. If disabled, D-Unit will not perform retraining on SR exit.
0	0h RW	**DQS Retrain Enable (DQS_RETRAIN_EN):** Periodic retraining enable: This bit enables periodic DQS retraining. If disabled, D-Unit will not perform retraining periodically. Note: Will be enabled only if DCO.IC is set and refreshes are enabled in DRF.MINREFRATE.

3.205 MR4 De-Swizzle Control (D_CR_MR4_DESWIZZLE) — Offset 1B84h

Controls the data bits swizzling crossbar for MR4.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 1B84h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD31):** Reserved
30:28	0h RW	**MR4 Bit 2 Select 2nd Byte (MR4_BIT2_SEL2):** Selects bit 2 of MR4 data.
27	0h RO	**Reserved (RSVD27):** Reserved
26:24	0h RW	**MR4 Bit 1 Select 2nd Byte (MR4_BIT1_SEL2):** Selects bit 1 of MR4 data
23	0h RO	**Reserved (RSVD23):** Reserved
22:20	0h RW	**MR4 Bit 0 Select 2nd Byte (MR4_BIT0_SEL2):** Selects bit 0 of MR4 data.
19:18	0h RO	**Reserved (RSVD19_18):** Reserved
17:16	0h RW	**MR4 Byte 2 Select (MR4_BYTE_SEL2):** Selects byte position of the MR4 data for second device.
15	0h RO	**Reserved (RSVD15):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
14:12	0h RW	**MR4 Bit 2 Select (MR4_BIT2_SEL):** Selects bit 2 of MR4 data.
11	0h RO	**Reserved (RSVD11):** Reserved
10:8	0h RW	**MR4 Bit 1 Select (MR4_BIT1_SEL):** Selects bit 1 of MR4 data.
7	0h RO	**Reserved (RSVD7):** Reserved
6:4	0h RW	**MR4 Bit 0 Select (MR4_BIT0_SEL):** Selects bit 0 of MR4 data.
3:2	0h RO	**Reserved (RSVD3_2):** Reserved
1:0	0h RW	**MR4 Byte Select (MR4_BYTE_SEL):** Selects byte position of the MR4 data first device.

3.206 Security Control Policy (C_CR_SECURITY_CP_0_0_0_MCHBAR) — Offset 6000h

Control policy register for the Security Policy Group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6000h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	0h RO	**Security Attribute of Initiator Permission Enable (SAI):** Each bit is associated with the SAI of an agent. When set to 1'b1, the associated agent is granted permission to update the CP, RAC, and WAC registers of the policy group.

3.207 Security Read Access Control Policy (C_CR_SECURITY_RAC_0_0_0_MCHBAR) — Offset 6008h

Read access control policy register for the Security Policy Group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6008h	C0061210202 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	C0061210202h RO	**Security Attribute of Initiator Permission Enable (SAI):** Each bit is associated with the SAI of an agent. When set to 1'b1, the associated agent is granted permission to read the registers contained in the policy group.

3.208 Security Write Access Control Policy (C_CR_SECURITY_WAC_0_0_0_MCHBAR) — Offset 6010h

Write access control policy register for the Security Policy Group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6010h	C0061210202 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	C0061210202h RO	**Security Attribute of Initiator Permission Enable (SAI):** Each bit is associated with the SAI of an agent. When set to 1'b1, the associated agent is granted permission to write the registers contained in the policy group.

3.209 C_CR_BIOSWR_CP (C_CR_BIOSWR_CP_0_0_0_MCHBAR) — Offset 6018h

Control policy register for the BIOS Write Policy Group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6018h	40001000202 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	40001000202h RW	**Security Attribute of Initiator Permission Enable (SAI):** Each bit is associated with the SAI of an agent. When set to 1'b1, the associated agent is granted permission to update the CP, RAC, and WAC registers of the policy group.

3.210 BIOSWR Read Access Control (C_CR_BIOSWR_RAC_0_0_0_MCHBAR) — Offset 6020h

Read access control policy register for the BIOS Write Policy Group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6020h	80000C0063010217 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	80000C0063010217h RW	**Security Attribute of Initiator Permission Enable (SAI):** Each bit is associated with the SAI of an agent. When set to 1'b1, the associated agent is granted permission to read the registers contained in the policy group.

3.211 C_CR_BIOSWR_WAC (C_CR_BIOSWR_WAC_0_0_0_MCHBAR) — Offset 6028h

Write access control policy register for the BIOS Write Policy Group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6028h	400010C0212 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	400010C0212h RW	**Security Attribute of Initiator Permission Enable (SAI):** Each bit is associated with the SAI of an agent. When set to 1'b1, the associated agent is granted permission to write the registers contained in the policy group.

3.212 P_U_CODEWR_ALLRD Control Policy (C_CR_P_U_CODEWR_ALLRD_CP_0_0_0_MCHBAR) — Offset 6030h

Control policy register for the P-Code/U-Code Write, All Read Policy Group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6030h	40001000202 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	40001000202h RW	**Security Attribute of Initiator Permission Enable (SAI):** Each bit is associated with the SAI of an agent. When set to 1'b1, the associated agent is granted permission to update the CP, RAC, and WAC registers of the policy group.

3.213 P_U_CODEWR_ALLRD Read Access Control Policy (C_CR_P_U_CODEWR_ALLRD_RAC_0_0_0_MCHBAR) — Offset 6038h

Read access control policy register for the P-Code/U-Code Write, All Read Policy Group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6038h	FFFFFFFFFFFFFFFF h

Bit Range	Default & Access	Field Name (ID): Description
63:0	FFFFFFFFFFFFFFFFh RO	**Security Attribute of Initiator Permission Enable (SAI):** Each bit is associated with the SAI of an agent. When set to 1'b1, the associated agent is granted permission to read the registers contained in the policy group.

3.214 P_U_CODEWR_ALLRD Write Access Control Policy (C_CR_P_U_CODEWR_ALLRD_WAC_0_0_0_MCHBAR) — Offset 6040h

Write access control policy register for the P-Code/U-Code Write, All Read Policy Group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6040h	40001000202 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	40001000202h RW	**Security Attribute of Initiator Permission Enable (SAI):** Each bit is associated with the SAI of an agent. When set to 1'b1, the associated agent is granted permission to write the registers contained in the policy group.

3.215 BIOS/PMC WR Control Policy (C_CR_BIOS_PMC_WR_CP_0_0_0_MCHBAR) — Offset 6050h

Control policy register for the BIOS/PMC Write Policy Group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6050h	C0061210202 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	C0061210202h RW	**Security Attribute of Initiator Permission Enable (SAI):** Each bit is associated with the SAI of an agent. When set to 1'b1, the associated agent is granted permission to update the CP, RAC, and WAC registers of the policy group.

3.216 BIOS/PMC WR Read Access Control (C_CR_BIOS_PMC_WR_RAC_0_0_0_MCHBAR) — Offset 6058h

Read access control policy register for the BIOS/PMC Write Policy Group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6058h	80000C0063210217 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	80000C0063210217h RW	**Security Attribute of Initiator Permission Enable (SAI):** Each bit is associated with the SAI of an agent. When set to 1'b1, the associated agent is granted permission to read the registers contained in the policy group.

3.217 BIOS/PMC WR Write Access Control (C_CR_BIOS_PMC_WR_WAC_0_0_0_MCHBAR) — Offset 6060h

Write access control policy register for the BIOS/PMC Write Policy Group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6060h	C00612C0212 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	C00612C0212h RW	**Security Attribute of Initiator Permission Enable (SAI):** Each bit is associated with the SAI of an agent. When set to 1'b1, the associated agent is granted permission to write the registers contained in the policy group.

3.218 P_U_PMC_CODEWR_ALLRD Control Policy (C_CR_P_U_PMC_CODEWR_ALLRD_CP_0_0_0_MCHBAR) — Offset 6070h

Control policy register for the P-Code/U-Code/PMC Write, All Read Policy Group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6070h	40001200202 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	40001200202h RW	**Security Attribute of Initiator Permission Enable (SAI):** Each bit is associated with the SAI of an agent. When set to 1'b1, the associated agent is granted permission to update the CP, RAC, and WAC registers of the policy group.

3.219 P_U_PMC_CODEWR_ALLRD Read Access Control Policy (C_CR_P_U_PMC_CODEWR_ALLRD_RAC_0_0_0_MCHBAR) — Offset 6078h

Read access control policy register for the P-Code/U-Code/PMC Write, All Read Policy Group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6078h	FFFFFFFFFFFFFFFF h

Bit Range	Default & Access	Field Name (ID): Description
63:0	FFFFFFFFFFFFFFFFh RO	**Security Attribute of Initiator Permission Enable (SAI):** Each bit is associated with the SAI of an agent. When set to 1'b1, the associated agent is granted permission to read the registers contained in the policy group.

3.220 P_U_PMC_CODEWR_ALLRD Write Access Control Policy (C_CR_P_U_PMC_CODEWR_ALLRD_WAC_0_0_0_MCHBAR) — Offset 6080h

Write access control policy register for the P-Code/U-Code/PMC Write, All Read Policy Group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6080h	40001200202 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	40001200202h RW	**Security Attribute of Initiator Permission Enable (SAI):** Each bit is associated with the SAI of an agent. When set to 1'b1, the associated agent is granted permission to write the registers contained in the policy group.

3.221 Upstream Device Arbiter Grant Count A2T (A_CR_UPARB_GCNT_DEV_A2T_MCHBAR) — Offset 6400h

Upstream Device arbiter grant count for A-Unit to T-Unit transactions.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6400h	1010101 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RW	**VC0B Completion Grant Count (VC0B_C):** VC0B completion grant count to T-Unit.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	1h RW	**VC0B Posted Grant Count (VC0B_P):** VC0B posted transaction grant count to T-Unit. This is only for MSIs.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**VC0A Completion Grant Count (VC0A_C):** VC0A completion grant count to T-Unit.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	1h RW	**VC0A Posted Grant Count (VC0A_P):** VC0A posted to T-Unit MSIs.

3.222 Upstream A2B Arbiter Channel 0 Grant Count (A_CR_UPARB_GCNT_A2B_0_MCHBAR) — Offset 6404h

Upstream Class/Target arbiter grant count.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6404h	1010101 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RW	**Channel ID 0 Target Arbiter Grant Count (CHID0_TGT):** Target arbiter grant count for the channel ID 0 class arbiter.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	1h RW	**Channel ID 0 Non-posted IOMMU Grant Count (CHID0_MMU_NP):** Grant count for IOMMU non-posted transactions on channel ID 0.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Channel ID 0 Non-posted Grant Count (CHID0_NP):** Grant count for non-posted transactions on channel ID 0.

Bit Range	Default & Access	Field Name (ID): Description
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	1h RW	**Channel ID 0 P Grant Count (CHID0_P):** Grant count for posted transactions on channel ID 0.

3.223 Upstream A2B Arbiter Channel 1 Grant Count (A_CR_UPARB_GCNT_A2B_1_MCHBAR) — Offset 6408h

Upstream Class/Target arbiter grant count.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6408h	1010101 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RW	**Channel ID 1 Target Arbiter Grant Count (CHID1_TGT):** Target arbiter grant count for the channel ID 1 class arbiter.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	1h RW	**Channel ID 1 Non-posted IOMMU Grant Count (CHID1_MMU_NP):** Grant count for IOMMU non-posted transactions on channel ID 1.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Channel ID 1 Non-posted Grant Count (CHID1_NP):** Grant count for non-posted transactions on channel ID 1.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	1h RW	**Channel ID 1 P Grant Count (CHID1_P):** Grant count for posted transactions on channel ID 1.

3.224 Upstream A2B Arbiter Channel 2 Grant Count (A_CR_UPARB_GCNT_A2B_2_MCHBAR) — Offset 640Ch

Upstream Class/Target arbiter grant count.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 640Ch	1000100 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RW	**Channel ID 2 Target Arbiter Grant Count (CHID2_TGT):** Target arbiter grant count for the channel ID 2 class arbiter.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	0h RO	**Channel ID 2 Non-posted IOMMU Grant Count (CHID2_MMU_NP):** Grant count for IOMMU non-posted transactions on channel ID 2.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Channel ID 2 Non-posted Grant Count (CHID2_NP):** Grant count for non-posted transactions on channel ID 2.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	0h RO	**Channel ID 2 P Grant Count (CHID2_P):** Grant count for posted transactions on channel ID 2.

3.225 Upstream A2B Arbiter Channel 3 Grant Count (A_CR_UPARB_GCNT_A2B_3_MCHBAR) — Offset 6410h

Upstream Class/Target arbiter grant count.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6410h	1010101 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RW	**Channel ID 3 Target Arbiter Grant Count (CHID3_TGT):** Target arbiter grant count for the channel ID 3 class arbiter.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	1h RW	**Channel ID 3 Non-posted IOMMU Grant Count (CHID3_MMU_NP):** Grant count for IOMMU non-posted transactions on channel ID 3.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Channel ID 3 Non-posted Grant Count (CHID3_NP):** Grant count for non-posted transactions on channel ID 3.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	1h RW	**Channel ID 3 P Grant Count (CHID3_P):** Grant count for posted transactions on channel ID 3.

3.226 Upstream A2B Arbiter Channel 4 Grant Count (A_CR_UPARB_GCNT_A2B_4_MCHBAR) — Offset 6414h

Upstream Class/Target arbiter grant count.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6414h	1000100 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RW	**Channel ID 4 Target Arbiter Grant Count (CHID4_TGT):** Target arbiter grant count for the channel ID 4 class arbiter.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	0h RO	**Channel ID 4 Non-posted IOMMU Grant Count (CHID4_MMU_NP):** Grant count for IOMMU non-posted transactions on channel ID 4.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Channel ID 4 Non-posted Grant Count (CHID4_NP):** Grant count for non-posted transactions on channel ID 4.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	0h RO	**Channel ID 4 P Grant Count (CHID4_P):** Grant count for posted transactions on channel ID 4.

3.227 Upstream A2B Arbiter Channel 5 Grant Count (A_CR_UPARB_GCNT_A2B_5_MCHBAR) — Offset 6418h

Upstream Class/Target arbiter grant count.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6418h	1000101 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RW	**Channel ID 5 Target Arbiter Grant Count (CHID5_TGT):** Target arbiter grant count for the channel ID 5 class arbiter.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	0h RO	**Channel ID 5 Non-posted IOMMU Grant Count (CHID5_MMU_NP):** Grant count for IOMMU non-posted transactions on channel ID 5.

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Channel ID 5 Non-posted Grant Count (CHID5_NP):** Grant count for non-posted transactions on channel ID 5.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	1h RW	**Channel ID 5 P Grant Count (CHID5_P):** Grant count for posted transactions on channel ID 5.

3.228 Upstream A2B Arbiter Channel 6 Grant Count (A_CR_UPARB_GCNT_A2B_6_MCHBAR) — Offset 641Ch

Upstream Class/Target arbiter grant count for A-Unit to B-Unit transactions.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 641Ch	1000101 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RW	**Channel ID 6 Target Arbiter Grant Count (CHID6_TGT):** Target arbiter grant count for the channel ID 6 class arbiter.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	0h RO	**Channel ID 6 Non-posted IOMMU Grant Count (CHID6_MMU_NP):** Grant count for IOMMU non-posted transactions on channel ID 6.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Channel ID 6 Non-posted Grant Count (CHID6_NP):** Grant count for non-posted transactions on channel ID 6.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	1h RW	**Channel ID 6 P Grant Count (CHID6_P):** Grant count for posted transactions on channel ID 6.

3.229 Upstream A2B Arbiter Channel 7 Grant Count (A_CR_UPARB_GCNT_A2B_7_MCHBAR) — Offset 6420h

Upstream Class/Target arbiter grant count for A-Unit to B-Unit transactions.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6420h	1000101 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RW	**Channel ID 7 Target Arbiter Grant Count (CHID7_TGT):** Target arbiter grant count for the channel ID 7 class arbiter.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	0h RO	**Channel ID 7 Non-posted IOMMU Grant Count (CHID7_MMU_NP):** Grant count for IOMMU non-posted transactions on channel ID 7.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Channel ID 7 Non-posted Grant Count (CHID7_NP):** Grant count for non-posted transactions on channel ID 7.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	1h RW	**Channel ID 7 P Grant Count (CHID7_P):** Grant count for posted transactions on channel ID 7.

3.230 Upstream A2T Arbiter Channel 0 Grant Count (A_CR_UPARB_GCNT_A2T_0_MCHBAR) — Offset 6424h

Upstream Class/Target arbiter grant count for A-Unit to T-Unit transactions.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6424h	1000101 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RW	**Channel ID 0 Target Arbiter Grant Count (CHID0_TGT):** Target arbiter grant count for the channel ID 0 a2t class arbiter.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	0h RO	**Channel ID 0 Non-posted IOMMU Grant Count (CHID0_MMU_NP):** Grant count for IOMMU non-posted transactions on channel ID 0.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Channel ID 0 Non-posted Grant Count (CHID0_NP):** Grant count for non-posted transactions on channel ID 0.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	1h RW	**Channel ID 0 P Grant Count (CHID0_P):** Grant count for posted transactions on channel ID 0.

3.231 Upstream P2P Arbiter Channel 0 Grant Count (A_CR_UPARB_GCNT_P2P_0_MCHBAR) — Offset 6428h

Upstream Class/Target arbiter grant count.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6428h	1010101 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RO	**Channel ID 0 Target Arbiter Grant Count (CHID0_TGT):** Target arbiter grant count for the channel ID 0 p2p class arbiter.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	1h RO	**Channel ID 0 C Grant Count (CHID0_C):** Grant count for completions on channel ID 0.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RO	**Channel ID 0 Non-posted Grant Count (CHID0_N):** Grant count for non-posted transactions on channel ID 0.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	1h RO	**Channel ID 0 P Grant Count (CHID0_P):** Grant count for posted transactions on channel ID 0.

3.232 Upstream P2P Arbiter Channel 1 Grant Count (A_CR_UPARB_GCNT_P2P_1_MCHBAR) — Offset 642Ch

Upstream Class/Target arbiter grant count.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 642Ch	1010101 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RO	**Channel ID 1 Target Arbiter Grant Count (CHID1_TGT):** Target arbiter grant count for the channel ID 1 p2p class arbiter.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	1h RO	**Channel ID 1 C Grant Count (CHID1_C):** Grant count for completions on channel ID 1.

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RO	**Channel ID 1 Non-posted Grant Count (CHID1_N):** Grant count for non-posted transactions on channel ID 1.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	1h RO	**Channel ID 1 P Grant Count (CHID1_P):** Grant count for posted transactions on channel ID 1

3.233 Upstream Private Credit Return Grant Count Posted 0 (A_CR_CRDARB_PRIV_GCNT_DEV_P_0_MCHBAR) — Offset 6430h

PSF credit return device arb grant count posted.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6430h	1000101 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RW	**Channel ID 3 P Grant Count (CHID3):** Grant count for posted transactions on channel ID 3.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	0h RO	**Channel ID 2 P Grant Count (CHID2):** Grant count for posted transactions on channel ID 2.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Channel ID 1 P Grant Count (CHID1):** Grant count for posted transactions on channel ID 1.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	1h RW	**Channel ID 0 P Grant Count (CHID0):** Grant count for posted transactions on channel ID 0.

3.234 Upstream Private Credit Return Grant Count Posted 1 (A_CR_CRDARB_PRIV_GCNT_DEV_P_1_MCHBAR) — Offset 6434h

PSF credit return device arb grant count posted.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6434h	1010100 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RW	**Channel ID 7 P Grant Count (CHID7):** Grant count for posted transactions on channel ID 7.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	1h RW	**Channel ID 6 P Grant Count (CHID6):** Grant count for posted transactions on channel ID 6.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Channel ID 5 P Grant Count (CHID5):** Grant count for posted transactions on channel ID 5.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	0h RO	**Channel ID 4 P Grant Count (CHID4):** Grant count for posted transactions on channel ID 4.

3.235 Upstream Private Credit Return Grant Count Non-posted 0 (A_CR_CRDARB_PRIV_GCNT_DEV_N_0_MCHBAR) — Offset 6438h

PSF credit return device arb grant count non-posted.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6438h	1010101 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RW	**Channel ID 3 Non-posted Grant Count (CHID3):** Grant count for non-posted transactions on channel ID 3.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	1h RW	**Channel ID 2 Non-posted Grant Count (CHID2):** Grant count for non-posted transactions on channel ID 2.
15:14	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:8	1h RW	**Channel ID 1 Non-posted Grant Count (CHID1):** Grant count for non-posted transactions on channel ID 1.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	1h RW	**Channel ID 0 Non-posted Grant Count (CHID0):** Grant count for non-posted transactions on channel ID 0.

3.236 Upstream Private Credit Return Grant Count Posted 1 (A_CR_CRDARB_PRIV_GCNT_DEV_N_1_MCHBAR) — Offset 643Ch

PSF credit return device arb grant count non-posted.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 643Ch	1040104 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RW	**Channel ID 7 Non-posted Grant Count (CHID7):** Grant count for non-posted transactions on channel ID 7.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	4h RW	**Channel ID 6 Non-posted Grant Count (CHID6):** Grant count for non-posted transactions on channel ID 6.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Channel ID 5 Non-posted Grant Count (CHID5):** Grant count for non-posted transactions on channel ID 5.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	4h RW	**Channel ID 4 Non-posted Grant Count (CHID4):** Grant count for non-posted transactions on channel ID 4.

3.237 Upstream Private Credit Return Grant Count Completion (A_CR_CRDARB_PRIV_GCNT_DEV_C_0_MCHBAR) — Offset 6440h

PSF credit return device arb grant count completion.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6440h	101 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RO	**Channel ID 3 C Grant Count (CHID3):** Grant count for completions on channel ID 3.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	0h RO	**Channel ID 2 C Grant Count (CHID2):** Grant count for completions on channel ID 2.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Channel ID 1 C Grant Count (CHID1):** Grant count for completions on channel ID 1.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	1h RW	**Channel ID 0 C Grant Count (CHID0):** Grant count for completions on channel ID 0.

3.238 Upstream Shared Credit Return Grant Count Posted 0 (A_CR_CRDARB_SHRD_GCNT_DEV_P_0_MCHBAR) — Offset 6444h

PSF credit return device arb grant count Posteds

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6444h	1000101 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RW	**Channel ID 3 P Grant Count (CHID3):** Grant count for posted transactions on channel ID 3.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	0h RO	**Channel ID 2 P Grant Count (CHID2):** Grant count for posted transactions on channel ID 2.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Channel ID 1 P Grant Count (CHID1):** Grant count for posted transactions on channel ID 1.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	1h RW	**Channel ID 0 P Grant Count (CHID0):** Grant count for posted transactions on channel ID 0.

3.239 Upstream Shared Credit Return Grant Count Posted 1 (A_CR_CRDARB_SHRD_GCNT_DEV_P_1_MCHBAR) — Offset 6448h

PSF credit return device arb grant count posted.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6448h	1010100 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RW	**Channel ID 7 P Grant Count (CHID7):** Grant count for posted transactions on channel ID 7.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	1h RW	**Channel ID 6 P Grant Count (CHID6):** Grant count for posted transactions on channel ID 6.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Channel ID 5 P Grant Count (CHID5):** Grant count for posted transactions on channel ID 5.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	0h RO	**Channel ID 4 P Grant Count (CHID4):** Grant count for posted transactions on channel ID 4.

3.240 Upstream Shared Credit Return Grant Count Non-posted 0 (A_CR_CRDARB_SHRD_GCNT_DEV_N_0_MCHBAR) — Offset 644Ch

PSF credit return device arb grant count non-posted.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 644Ch	1010101 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RW	**Channel ID 3 Non-posted Grant Count (CHID3):** Grant count for non-posted transactions on channel ID 3.
23:22	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
21:16	1h RW	**Channel ID 2 Non-posted Grant Count (CHID2):** Grant count for non-posted transactions on channel ID 2.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Channel ID 1 Non-posted Grant Count (CHID1):** Grant count for non-posted transactions on channel ID 1.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	1h RW	**Channel ID 0 Non-posted Grant Count (CHID0):** Grant count for non-posted transactions on channel ID 0.

3.241 Upstream Shared Credit Return Grant Count Posted 1 (A_CR_CRDARB_SHRD_GCNT_DEV_N_1_MCHBAR) — Offset 6450h

PSF credit return device arb grant count non-posted.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6450h	1040104 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	1h RW	**Channel ID 7 Non-posted Grant Count (CHID7):** Grant count for non-posted transactions on channel ID 7.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	4h RW	**Channel ID 6 Non-posted Grant Count (CHID6):** Grant count for non-posted transactions on channel ID 6.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Channel ID 5 Non-posted Grant Count (CHID5):** Grant count for non-posted transactions on channel ID 5.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	4h RW	**Channel ID 4 Non-posted Grant Count (CHID4):** Grant count for non-posted transactions on channel ID 4.

3.242 Upstream Shared Credit Return Grant Count Completion (A_CR_CRDARB_SHRD_GCNT_DEV_C_0_MCHBAR) — Offset 6454h

PSF credit return device arb grant count completion.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6454h	101 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RO	**Channel ID 3 C Grant Count (CHID3):** Grant count for completions on channel ID 3.
23:22	0h RO	**Reserved (RSVD):** Reserved
21:16	0h RO	**Channel ID 2 C Grant Count (CHID2):** Grant count for completions on channel ID 2.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Channel ID 1 C Grant Count (CHID1):** Grant count for completions on channel ID 1.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	1h RW	**Channel ID 0 C Grant Count (CHID0):** Grant count for completions on channel ID 0.

3.243 Upstream Credit Arbiter Private Credit Return Class Arbiter Grant Count (A_CR_CRDARB_PRIV_GCNT_CLS_MCHBAR) — Offset 6458h

PSF credit return class arb grant count

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6458h	10101 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (RSVD):** Reserved
21:16	1h RW	**C:** All Completions.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Non-posted (NP):** All non-posted.
7:6	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5:0	1h RW	**P:** All posted.

3.244 Upstream Credit Arbiter Shared Cedit Return Class Arbiter Grant Count (A_CR_CRDARB_SHRD_GCNT_CLS_MCHBAR) — Offset 645Ch

PSF credit return class arb grant count

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 645Ch	10101 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (RSVD):** Reserved
21:16	1h RW	**C:** All Completions
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**NP:** All Non Posteds
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	1h RW	**P:** All Posteds

3.245 Gazelle Queue Limit Channel 0-3 (A_CR_GZLQ_LIMIT_CH0_3_MCHBAR) — Offset 6460h

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6460h	FFFFFFFF h

Bit Range	Default & Access	Field Name (ID): Description
31:24	FFh RW	**Channel ID 3 (CHID3):** Gazelle queue limit for Upstream NP on channel ID 3.
23:16	FFh RW	**Channel ID 2 (CHID2):** Gazelle queue limit for Upstream NP on channel ID 2.
15:8	FFh RW	**Channel ID 1 (CHID1):** Gazelle queue limit for Upstream NP on channel ID 1.

Bit Range	Default & Access	Field Name (ID): Description
7:0	FFh RW	**Channel ID 0 (CHID0):** Gazelle queue limit for Upstream NP on channel ID 0.

3.246 Gazelle Queue Limit Channels 4-7 (A_CR_GZLQ_LIMIT_CH4_7_MCHBAR) — Offset 6464h

Gazelle queue limit for channel ID 4 to 7.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6464h	FFFFFFFF h

Bit Range	Default & Access	Field Name (ID): Description
31:24	FFh RW	**Channel ID 7 (CHID7):** Gazelle queue limit for Upstream NP on channel ID 7.
23:16	FFh RW	**Channel ID 6 (CHID6):** Gazelle queue limit for Upstream NP on channel ID 6.
15:8	FFh RW	**Channel ID 5 (CHID5):** Gazelle queue limit for Upstream NP on channel ID 5.
7:0	FFh RW	**Channel ID 4 (CHID4):** Gazelle queue limit for Upstream NP on channel ID 4.

3.247 IOMMU Arbiter Grant Count VC0a Register (A_CR_IOMMUARB_GCNT_VC0A_0_0_0_MCHBAR) — Offset 6468h

IOMMU arbiter grant count.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6468h	10101 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (RSVD):** Reserved
21:16	1h RW	**Class Count (CLASS_CNT):** IOMMU arbiter class grant count.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Device Non-posted (DEV_NP):** IOMMU arbiter device grant count for NP.
7:6	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5:0	1h RW	**Device Posted (DEV_P):** IOMMU arbiter device grant count for p.

3.248 IOMMU Arbiter Grant Count VC0b Register (A_CR_IOMMUARB_GCNT_VC0B_0_0_0_MCHBAR) — Offset 646Ch

IOMMU arbiter grant count.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 646Ch	10101 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (RSVD):** Reserved
21:16	1h RW	**Class Count (CLASS_CNT):** IOMMU arbiter class grant count.
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Device Non-posted (DEV_NP):** IOMMU arbiter device grant count for NP.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	1h RW	**Device Posted (DEV_P):** IOMMU arbiter device grant count for p.

3.249 IOMMU Arbiter Grant Count VC1b Register (A_CR_IOMMUARB_GCNT_VC1B_0_0_0_MCHBAR) — Offset 6470h

IOMMU arbiter grant count.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6470h	10101 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (RSVD):** Reserved
21:16	1h RW	**Class Count (CLASS_CNT):** IOMMU arbiter class grant count.

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RO	**Reserved (RSVD):** Reserved
13:8	1h RW	**Device Non-posted (DEV_NP):** IOMMU arbiter device grant count for NP.
7:6	0h RO	**Reserved (RSVD):** Reserved
5:0	1h RW	**Device Posted (DEV_P):** IOMMU arbiter device grant count for p.

3.250 Gazelle Queue Reserved Entries Channels 0-3 (A_CR_GZLQ_RSVD_CH0_3_MCHBAR) — Offset 6474h

Gazelle queue limit for channel ID 0 to 3.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6474h	4040101 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	4h RW	**Channel ID 3 (CHID3):** Gazelle queue limit for Upstream NP on channel ID 3.
23:16	4h RW	**Channel ID 2 (CHID2):** Gazelle queue limit for Upstream NP on channel ID 2.
15:8	1h RW	**Channel ID 1 (CHID1):** Gazelle queue limit for Upstream NP on Channel ID 1.
7:0	1h RW	**Channel ID 0 (CHID0):** Gazelle queue limit for Upstream NP on Channel ID 0.

3.251 Gazelle Queue Reserved Entries Channels 4-7 (A_CR_GZLQ_RSVD_CH4_7_MCHBAR) — Offset 6478h

GazelleQ limit for CHID 0 to 3

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6478h	1010101 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	1h RW	**Channel ID 7 (CHID7):** Gazelle queue limit for Upstream NP on channel ID 7.
23:16	1h RW	**Channel ID 6 (CHID6):** Gazellequeue limit for Upstream NP on channel ID 6.

Bit Range	Default & Access	Field Name (ID): Description
15:8	1h RW	**Channel ID 5 (CHID5):** Gazelle queue limit for Upstream NP on channel ID 5.
7:0	1h RW	**Channel ID 4 (CHID4):** Gazelle queue limit for Upstream NP on channel ID 4.

3.252 Spare BIOS (A_CR_SPARE_BIOS_MCHBAR) — Offset 647Ch

Spare CR in MCHBAR.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 647Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SPARE RW Bits (SPARE_RW):** Spare RW 32 bits in BIOSWR policy group.

3.253 Upcmd Credit Maximum Channel 0 (A_CR_UPCMD_CRDTMAX_CH0_0_0_0_MCHBAR) — Offset 6490h

Upcmd Credit Max for PSF0 on Ch0 : Maximum number of credits exposed to PSF0.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6490h	4040C h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RO	**Reserved (RSVD):** Reserved
22:16	4h RW	**Completion Max (CMP_MAX):** Max Cmp credits sent to PSF0 for Chid.
15	0h RO	**Reserved (RSVD):** Reserved
14:8	4h RW	**Non-Posted Max (NP_MAX):** Max non-posted credits sent to PSF0 for Chid.
7	0h RO	**Reserved (RSVD):** Reserved
6:0	Ch RW	**Posted Max (P_MAX):** Max posted credits sent to PSF0 for Chid.

3.254 Upcmd Credit Maximum Channel 1 (A_CR_UPCMD_CRDTMAX_CH1_0_0_0_MCHBAR) — Offset 6494h

Upcmd Credit Max for PSF0 on Ch1 : Maximum number of credits exposed to PSF0.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6494h	40404 h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RO	**Reserved (RSVD):** Reserved
22:16	4h RW	**Completion Max (CMP_MAX):** Max Cmp credits sent to PSF0 for Chid.
15	0h RO	**Reserved (RSVD):** Reserved
14:8	4h RW	**Non-Posted Max (NP_MAX):** Max non-posted credits sent to PSF0 for Chid.
7	0h RO	**Reserved (RSVD):** Reserved
6:0	4h RW	**Posted Max (P_MAX):** Max posted credits sent to PSF0 for Chid.

3.255 Upcmd Credit Maximum Channel 2 (A_CR_UPCMD_CRDTMAX_CH2_0_0_0_MCHBAR) — Offset 6498h

Upcmd Credit Max for PSF0 on Ch2 : Maximum number of credits exposed to PSF0.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6498h	7F00 h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RO	**Reserved (RSVD):** Reserved
22:16	0h RW	**Completion Max (CMP_MAX):** Max Cmp credits sent to PSF0 for Chid.
15	0h RO	**Reserved (RSVD):** Reserved
14:8	7Fh RW	**Non-Posted Max (NP_MAX):** Max non-posted credits sent to PSF0 for Chid.
7	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
6:0	0h RW	**Posted Max (P_MAX):** Max posted credits sent to PSF0 for Chid.

3.256 Upcmd Credit Maximum Channel 3 (A_CR_UPCMD_CRDTMAX_CH3_0_0_0_MCHBAR) — Offset 649Ch

Upcmd Credit Max for PSF0 on Ch3 : Maximum number of credits exposed to PSF0.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 649Ch	7F7F h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RO	**Reserved (RSVD):** Reserved
22:16	0h RW	**Completion Max (CMP_MAX):** Max Cmp credits sent to PSF0 for Chid.
15	0h RO	**Reserved (RSVD):** Reserved
14:8	7Fh RW	**Non-Posted Max (NP_MAX):** Max non-posted credits sent to PSF0 for Chid.
7	0h RO	**Reserved (RSVD):** Reserved
6:0	7Fh RW	**Posted Max (P_MAX):** Max posted credits sent to PSF0 for Chid.

3.257 Upcmd Credit Maximum Channel 4 (A_CR_UPCMD_CRDTMAX_CH4_0_0_0_MCHBAR) — Offset 64A0h

Upcmd Credit Max for PSF0 on Ch4 : Maximum number of credits exposed to PSF0.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 64A0h	400 h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RO	**Reserved (RSVD):** Reserved
22:16	0h RW	**Completion Max (CMP_MAX):** Max Cmp credits sent to PSF0 for Chid.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**Reserved (RSVD):** Reserved
14:8	4h RW	**Non-Posted Max (NP_MAX):** Max non-posted credits sent to PSF0 for Chid.
7	0h RO	**Reserved (RSVD):** Reserved
6:0	0h RW	**Posted Max (P_MAX):** Max posted credits sent to PSF0 for Chid.

3.258 Upcmd Credit Maximum Channel 5 (A_CR_UPCMD_CRDTMAX_CH5_0_0_0_MCHBAR) — Offset 64A4h

Upcmd Credit Max for PSF0 on Ch5 : Maximum number of credits exposed to PSF0.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 64A4h	404 h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RO	**Reserved (RSVD):** Reserved
22:16	0h RW	**Completion Max (CMP_MAX):** Max Cmp credits sent to PSF0 for Chid.
15	0h RO	**Reserved (RSVD):** Reserved
14:8	4h RW	**Non-Posted Max (NP_MAX):** Max non-posted credits sent to PSF0 for Chid.
7	0h RO	**Reserved (RSVD):** Reserved
6:0	4h RW	**Posted Max (P_MAX):** Max posted credits sent to PSF0 for Chid.

3.259 Upcmd Credit Maximum Channel 6 (A_CR_UPCMD_CRDTMAX_CH6_0_0_0_MCHBAR) — Offset 64A8h

Upcmd Credit Max for PSF0 on Ch6 : Maximum number of credits exposed to PSF0.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 64A8h	404 h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RO	**Reserved (RSVD):** Reserved
22:16	0h RW	**Completion Max (CMP_MAX):** Max Cmp credits sent to PSF0 for Chid.
15	0h RO	**Reserved (RSVD):** Reserved
14:8	4h RW	**Non-Posted Max (NP_MAX):** Max non-posted credits sent to PSF0 for Chid.
7	0h RO	**Reserved (RSVD):** Reserved
6:0	4h RW	**Posted Max (P_MAX):** Max posted credits sent to PSF0 for Chid.

3.260 Upcmd Credit Maximum Channel 7 (A_CR_UPCMD_CRDTMAX_CH7_0_0_0_MCHBAR) — Offset 64ACh

Upcmd Credit Max for PSF0 on Ch7 : Maximum number of credits exposed to PSF0.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 64ACh	40C h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RO	**Reserved (RSVD):** Reserved
22:16	0h RW	**Completion Max (CMP_MAX):** Max Cmp credits sent to PSF0 for Chid.
15	0h RO	**Reserved (RSVD):** Reserved
14:8	4h RW	**Non-Posted Max (NP_MAX):** Max non-posted credits sent to PSF0 for Chid.
7	0h RO	**Reserved (RSVD):** Reserved
6:0	Ch RW	**Posted Max (P_MAX):** Max posted credits sent to PSF0 for Chid.

3.261 MOT OUT Base Register (A_CR_MOT_OUT_BASE_0_0_0_MCHBAR) — Offset 64C0h

This register contains the value of the start address of the MOT debug data region. The smallest reserved region for MOT debug data (if enabled) is 16MB. MOT region must be power-of-two sized and naturally

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 64C0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR enable (IMR_EN):** IMR Enable: Enables access checking for the MOT region. Note: this does not enable MOT itself merely enables access control checks for transactions that attempt to access the MOT buffer.
30	0h RO	**Trace Enable (TR_EN):** Asset Classification (AC)[0]: Trace Enable: Enables snooping of transactions to the IMR region by tracing agents such as MOT. Reserved and set to 0 for the MOT region, since otherwise this would enable recursive
29	0h RO	**Reserved (RSVD):** Reserved
28:14	0h RW	**MOT_OUT_BASE:** Specifies bits 38:24 of the start address of the MOT memory region. Region size must be a strict poweroftwo at least 16MB and naturally aligned to the size. These bits are compared with the result of the MOT_OUT_MASK[28:14] applied to bits 38:24 of the incoming address to determine if an access falls within the MOT region.
13:0	0h RO	**Reserved (RSVD):** Reserved

3.262 MOT OUT Mask Register (A_CR_MOT_OUT_MASK_0_0_0_MCHBAR) — Offset 64C4h

This register specifies the size of the MOT region. If a request address [39:24] AND-ed with MOT_OUT_MASK[15:0] matches the MOT_OUT_BASE[15:0], then the request falls within the MOT_OUT region

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 64C4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GT Implicit Writeback Enable (GT_IWB_EN):** Asset Classification AC[2]: GT Implicit WB Enable: Enables implicit writebacks to protected region from GT caching agent. When set to 1 enables implicit writeback data HITM data from GT to be returned to the requester. When set to 0 inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**IA Implicit Writeback Enable (IA_IWB_EN):** Asset Classification AC[1]: IA Implicit WB Enable: Enables implicit writebacks to protected region from IA caching agent. When set to 1 enables implicit writeback data HITM data from IA cores to be returned to the requester. When set to 0 inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RSVD):** Reserved
28:14	0h RW	**MOT OUT Mask (MOT_OUT_MASK):** Specifies the size of the MOT region. If Request Address [38:24] ANDed with MOT_OUT_MASK[28:14] matches the MOT_OUT_BASE[28:14] then the request falls within the MOT_OUT region

Bit Range	Default & Access	Field Name (ID): Description
13:0	0h RO	**Reserved (RSVD):** Reserved

3.263 A-Unit BIOSWR Control Policy (A_CR_BIOSWR_CP_0_0_0_MCHBAR) — Offset 64C8h

This register controls the access policy to the A-Unit BIOS_RAC BIOS_AC BIOS_CP.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 64C8h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	C0061010202h RW	**BIOS Control Policy (BIOSWR_CP):** Bit vector used to determine which agents are allowed access to the A-Unit BIOSWR Registers based on the value from the agents 6 bit SAI field. This register is selfreferential the access policy provided applies to access to the control register itself.

3.264 A-Unit BIOSWR Ready Access Control (A_CR_BIOSWR_RAC_0_0_0_MCHBAR) — Offset 64D0h

This register configures the Read Access Policy for the A-Unit BIOS policy configuration registers. It is programmed with a SAI Policy that indicates which agents in the system are allowed to perform read operations to A-Unit BIOS policy group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 64D0h	80000C0063010217 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	80000C0063010217h RW	**BIOS Read Access Control (BIOSWR_RAC):** Bit vector used to determine which agents are allowed Rd access to the A-Unit BIOSWR Register based on the value from the agents 6 bit SAI field.

3.265 BIOSWR Write Access Control (A_CR_BIOSWR_WAC_0_0_0_MCHBAR) — Offset 64D8h

This register configures the Write Access Policy for the A-Unit BIOSWR registers. It is programmed with a SAI Policy that indicates which agents in the system are allowed to perform read operations to A-Unit BIOS policy group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 64D8h	C0061000212 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	C0061000212h RW	**BIOS Write Access Control (BIOSWR_WAC):** Bit vector used to determine which agents are allowed Wr access to the A-Unit BIOSWR registers based on the value from the agents 6 bit SAI field.

3.266 AUnit Pcode/Ucode Write, All Read Control Policy Register (A_CR_P_U_CODEWR_ALLRD_CP_0_0_0_MCHBAR) — Offset 64E0h

AUnit Pcode/Ucode Write, All Read Control Policy: This register controls the access policy to the Aunit P_U_CODEWR_ALLRD_RAC P_U_CODEWR_ALLRD_WAC

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 64E0h	40001000202 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	40001000202h RW	**P_U_CODEWR_ALLRD Control Policy (P_U_CODEWR_ALLRD_CP):** Bit vector used to determine which agents are allowed access to the Aunit P_U_CODEWR_ALLRD based on the value from the agents 6 bit SAI field. This register is selfreferential the access policy provided applies to access to the control register itself.

3.267 AUnit Pcode/Ucode Write, All Read Read Access Control Policy Register (A_CR_P_U_CODEWR_ALLRD_RAC_0_0_0_MCHBAR) — Offset 64E8h

AUnit Pcode/Ucode Write, All Read Read Access Control Policy: This register controls the read access policy to the Aunit P_U_CODEWR_ALLRD

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 64E8h	FFFFFFFFFFFFFFFF h

Bit Range	Default & Access	Field Name (ID): Description
63:0	FFFFFFFFFFFFFFFFh RO	**P_U_CODEWR_ALLRD Read Access Control (P_U_CODEWR_ALLRD_RAC):** Bit vector used to determine which agents are allowed Rd access to the Aunit P_U_CODEWR_ALLRD based on the value from the agents 6 bit SAI field.

3.268 AUnit Pcode/Ucode Write, All Read Write Access Control Policy Register (A_CR_P_U_CODEWR_ALLRD_WAC_0_0_0_MCHBAR) — Offset 64F0h

AUnit Pcode/Ucode Write, All Read Write Access Control Policy: This register controls the write access policy to the Aunit P_U_CODEWR_ALLRD

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 64F0h	40001000202 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	40001000202h RW	**P_U_CODEWR_ALLRD Write Access Control (P_U_CODEWR_ALLRD_WAC):** Bit vector used to determine which agents are allowed Wr access to the Aunit P_U_CODEWR_ALLRD based on the value from the agents 6 bit SAI field.

3.269 CHAP Select 1 (A_CR_CHAP_SLCT1_MCHBAR) — Offset 6500h

Chap event select register 1.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6500h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23:16	0h RW	**Channel ID Y Count Upstream txn (CHID_Y_CNT_UPTXN):** Count upstream txn on all VCs where CHID_Y[i]=1.
15:8	0h RO	**Reserved (RSVD):** Reserved
7:0	0h RW	**Channel ID X Count Upstream txn (CHID_X_CNT_UPTXN):** Count upstream txn on all VCs where CHID_X[i]=1.

3.270 CHAP Select 2 (A_CR_CHAP_SLCT2_MCHBAR) — Offset 6504h

Chap event select register 2.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6504h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23:16	0h RW	**Channel ID Y Count Downstream txn (CHID_Y_CNT_DNTXN):** Count downstream txn on all VCs where CHID_Y[i]=1.
15:8	0h RO	**Reserved (RSVD):** Reserved
7:0	0h RW	**Channel ID X Count Downstream txn (CHID_X_CNT_DNTXN):** Count downstream txn on all VCs where CHID_X[i]=1.

3.271 CHAP Select 3 (A_CR_CHAP_SLCT3_MCHBAR) — Offset 6508h

Chap event select register 3.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6508h	4 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RSVD):** Reserved
3:2	1h RW	**Q Occupancy Y (Q_OCCUPANCY_Y):** Count occupancy/residency of certain Q in A-Unit • 00: UpCmd • 01: UpData • 10: GzlQ • 11: DnCmd/Data
1:0	0h RW	**Q Occupancy X (Q_OCCUPANCY_X):** Count occupancy/residency of certain Q in A-Unit • 00: UpCmd • 01: UpData • 10: GzlQ • 11: DnCmd/Data

3.272 A_IMRGLOBAL_BM Control Policy (A_CR_IMRGLOBAL_BM_CP_0_0_0_MCHBAR) — Offset 6510h

This register controls the access policy to the A-Unit IMRGLOBAL_BM_RAC IMRGLOBAL_BM_AC IMRGLOBAL_BM_CP.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6510h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	C0061010202h RW	**IMRGLOBAL_BM Control Policy (IMRGLOBAL_BM_CP):** Bit vector used to determine which agents are allowed access to the A-Unit IMRGLOBAL_BM Registers based on the value from the agents 6 bit SAI field. This register is selfreferential the access policy provided applies to access to the control register itself.

3.273 A_IMRGLOBAL_BM Read Access Control (A_CR_IMRGLOBAL_BM_RAC_0_0_0_MCHBAR) — Offset 6518h

This register configures the Read Access Policy for the A-Unit IMRGLOBAL_BM policy configuration registers. It is programmed with a SAI Policy that indicates which agents in the system are allowed to perform read operations to A-Unit IMRGLOBAL_BM policy group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6518h	FFFFFFFFFFFFFFFF h

Bit Range	Default & Access	Field Name (ID): Description
63:0	FFFFFFFFFFFFFFFFh RO	**IMRGLOBAL_BM Read Access Control (IMRGLOBAL_BM_RAC):** Bit vector used to determine which agents are allowed Rd access to the A-Unit IMRGLOBAL_BM Register based on the value from the agents 6 bit SAI field.

3.274 A_IMRGLOBAL_BM Write Access Control (A_CR_IMRGLOBAL_BM_WAC_0_0_0_MCHBAR) — Offset 6520h

This register configures the Write Access Policy for the A-Unit IMRGLOBAL_BM registers. It is programmed with a SAI Policy that indicates which agents in the system are allowed to perform read operations to A-Unit IMRGLOBAL_BM policy group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6520h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	C0061010202h RW	**IMRGLOBAL_BM Write Access Control (IMRGLOBAL_BM_WAC):** Bit vector used to determine which agents are allowed Wr access to the A-Unit IMRGLOBAL_BM registers based on the value from the agents 6 bit SAI field.

3.275 Uncorrectable Error Status Register (A_CR_UNCERRSTS_0_0_0_MCHBAR) — Offset 6588h

Errors that have been seen in aunit. The error is only logged if the respective bit in UNCERRMSK is 0.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6588h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**reserved (UNDEFINED31):** reserved for future use
30	0h RO	**reserved (UNDEFINED30):** reserved for future use
29	0h RO	**reserved (UNDEFINED29):** reserved for future use
28	0h RO	**reserved (UNDEFINED28):** reserved for future use
27	0h RO	**reserved (UNDEFINED27):** reserved for future use
26	0h RO	**reserved (UNDEFINED26):** reserved for future use
25	0h RO	**reserved (UNDEFINED25):** reserved for future use
24	0h RO	**reserved (UNDEFINED24):** reserved for future use
23	0h RO	**reserved (UNDEFINED23):** reserved for future use
22	0h RO	**reserved (UNDEFINED22):** reserved for future use
21	0h RO	**reserved (UNDEFINED21):** reserved for future use

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**reserved (UNDEFINED20):** reserved for future use
19	0h RO	**reserved (UNDEFINED19):** reserved for future use
18	0h RO	**reserved (UNDEFINED18):** reserved for future use
17	0h RO	**reserved (UNDEFINED17):** reserved for future use
16	0h RO	**reserved (UNDEFINED16):** reserved for future use
15	0h RO	**reserved (UNDEFINED15):** reserved for future use
14	0h RO	**reserved (UNDEFINED14):** reserved for future use
13	0h RO	**reserved (UNDEFINED13):** reserved for future use
12	0h RO	**reserved (UNDEFINED12):** reserved for future use
11	0h RO	**reserved (UNDEFINED11):** reserved for future use
10	0h RW/1C	**msi rsvd set (MSI_RSVD_SET):** An MSI was received with reserved bits set
9	0h RW/1C	**gpa overflow (GPA_OVERFLOW):** A transaction was received with a guest physical address that was too large
8	0h RW/1C	**illegal msi (ILLEGAL_MSI):** A malformed/illegal MSI was received in the upstream direction
7	0h RW/1C	**at translated illegal device (AT_TRANSLATED_ILLEGAL_DEVICE):** A device that is not support to set the AT bit set it to an illegal value
6	0h RO	**reserved (UNDEFINED5):** reserved for future use
5	0h RW/1C	**bad sai cmpl (BAD_SAI_CMPL):** An incorrect/illegal sai was received with an upstream completion transaction.
4	0h RW/1C	**received lk cmpl (RECEIVED_LK_CMPL):** Received a CmplLck completion from iosf.
3	0h RW/1C	**bad sai nonposted (BAD_SAI_NONPOSTED):** An incorrect/illegal sai was received with an upstream non-posted transaction.
2	0h RW/1C	**illegal nonposted opcode (ILLEGAL_NONPOSTED_OPCODE):** Illegal/ Unsupported non-posted opcode received from iosf.
1	0h RW/1C	**bad sai posted (BAD_SAI_POSTED):** An incorrect/illegal sai was received with an upstream posted transaction.
0	0h RW/1C	**illegal posted opcode (ILLEGAL_POSTED_OPCODE):** Illegal/Unsupported posted opcode received from iosf.

3.276 Uncorrectable Error Mask Register (A_CR_UNCERRMSK_0_0_0_MCHBAR) — Offset 658Ch

Masks whether a reported error is logged and signaled to the IEH. 1- do not log/signal. 0 - log/signal.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 658Ch	FFFFFFFF h

Bit Range	Default & Access	Field Name (ID): Description
31	1h RO	**reserved (UNDEFINED31):** reserved for future use
30	1h RO	**reserved (UNDEFINED30):** reserved for future use
29	1h RO	**reserved (UNDEFINED29):** reserved for future use
28	1h RO	**reserved (UNDEFINED28):** reserved for future use
27	1h RO	**reserved (UNDEFINED27):** reserved for future use
26	1h RO	**reserved (UNDEFINED26):** reserved for future use
25	1h RO	**reserved (UNDEFINED25):** reserved for future use
24	1h RO	**reserved (UNDEFINED24):** reserved for future use
23	1h RO	**reserved (UNDEFINED23):** reserved for future use
22	1h RO	**reserved (UNDEFINED22):** reserved for future use
21	1h RO	**reserved (UNDEFINED21):** reserved for future use
20	1h RO	**reserved (UNDEFINED20):** reserved for future use
19	1h RO	**reserved (UNDEFINED19):** reserved for future use
18	1h RO	**reserved (UNDEFINED18):** reserved for future use
17	1h RO	**reserved (UNDEFINED17):** reserved for future use
16	1h RO	**reserved (UNDEFINED16):** reserved for future use
15	1h RO	**reserved (UNDEFINED15):** reserved for future use
14	1h RO	**reserved (UNDEFINED14):** reserved for future use

Bit Range	Default & Access	Field Name (ID): Description
13	1h RO	**reserved (UNDEFINED13):** reserved for future use
12	1h RO	**reserved (UNDEFINED12):** reserved for future use
11	1h RO	**reserved (UNDEFINED11):** reserved for future use
10	1h RW	**msi rsvd set (MSI_RSVD_SET):** An MSI was received with reserved bits set
9	1h RW	**gpa overflow (GPA_OVERFLOW):** A transaction was received with a guest physical address that was too large
8	1h RW	**illegal msi (ILLEGAL_MSI):** A malformed/illegal MSI was received in the upstream direction
7	1h RW	**at translated illegal device (AT_TRANSLATED_ILLEGAL_DEVICE):** A device that is not support to set the AT bit set it to an illegal value
6	1h RO	**reserved (UNDEFINED5):** reserved for future use
5	1h RW	**bad sai cmpl (BAD_SAI_CMPL):** An incorrect/illegal sai was received with an upstream completion transaction.
4	1h RW	**received lk cmpl (RECEIVED_LK_CMPL):** Received a CmplLck completion from iosf.
3	1h RW	**bad sai nonposted (BAD_SAI_NONPOSTED):** An incorrect/illegal sai was received with an upstream non-posted transaction.
2	1h RW	**illegal nonposted opcode (ILLEGAL_NONPOSTED_OPCODE):** Illegal/ Unsupported non-posted opcode received from iosf.
1	1h RW	**bad sai posted (BAD_SAI_POSTED):** An incorrect/illegal sai was received with an upstream posted transaction.
0	1h RW	**illegal posted opcode (ILLEGAL_POSTED_OPCODE):** Illegal/Unsupported posted opcode received from iosf.

3.277 Slice and Channel Hash (A_CR_SLICE_CHANNEL_HASH_0_0_0_MCHBAR) — Offset 65C0h

A-Unit slice and channel hash function.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 65C0h	3C00000000h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**LOCK:** Intended usage is for BIOS to set the LOCK when it updates the CR. A-Unit implements only storage for this bit. No hardware exists to implement hardware locking. NOTE: B-Unit copy of this bit is used by ucode to protect WRMSR to the B-Unit's CR. Ucode has no access to the A-Unit's copy of the CR.

Bit Range	Default & Access	Field Name (ID): Description
62:52	0h RO	**Reserved (RSVD):** Reserved
51:38	0h RW	**Channel Hash Mask (CH_HASH_MASK):** When both PMI channels in a slice are enabled, this field specifies the Channel Hash Mask to be applied on Addr[19:6] postremap DRAM address of the request to compute which PMI channel a request must be routed to. Relevant only when HVM mode is disabled and only for requests that do not fall under the MOT region. B-Unit will override the programmed value to include the Channel Selector bit See SLICEHASH.INTERLEAVE_MODE field. Note that HVM mode and MOT regions have special hash requirements and hence they do not use the CH_HASH_MASK.
37:36	3h RW	**Channel Enabled for Slice 1 (SYM_SLICE1_CHANNEL_ENABLED):** Specifies which channel is enabled for Slice 1, This is for those cases where Channel 0 or 1 could be disabled for Slice 1 If both bits are set then channel select will be based on the Channel Select logic
35:34	3h RW	**Channel Enabled for Slice 0 (SYM_SLICE0_CHANNEL_ENABLED):** Specifies which channel is enabled for Slice 0, This is for those cases where Channel 0 or 1 could be disabled for Slice 0 If both bits are set then channel select will be based on the Channel Select logic
33	0h RO	**Reserved (RSVD):** Reserved
32	0h RW	**Channel 1 Disabled (CH_1_DISABLED):** Channel 1 in both slices are disabled no memory address mapped to ch 1. All requests sent to channel 0.
31	0h RW	**Enable PMI Dual Data Mode (ENABLE_PMI_DUAL_DATA_MODE):** When set to 1, Single Command Interface and Dual Data Interface for Reads and Writes
30:20	0h RO	**Reserved (RSVD):** Reserved
19:6	0h RW	**Slice Hash Mask (SLICE_HASH_MASK):** When both slices are enabled this field specifies the Slice Hash Mask to be applied on Addr[19:6] physical address of the request to compute which slice a request must be routed to. Relevant only when HVM mode is disabled and only for physical addresses that do not fall under the Asymmetric Memory Region and the MOT region. B-Unit will override the programmed value to include the Slice Selector bit See INTERLEAVE_MODE field. Note that HVM mode nonaddress IDI requests asymmetric memory region and MOT regions have special hash requirements and hence they do not use the SLICE_HASH_MASK.
5	0h RO	**Reserved (RSVD):** Reserved
4	0h RW	**Slice 0 Mem Disabled (SLICE_0_MEM_DISABLED):** Slice 0 is disabled for memory accesses; no memory address mapped to Slice 0 and all memory requests sent to Slice 1.

Bit Range	Default & Access	Field Name (ID): Description
3:2	0h RW	**Interleave Mode (INTERLEAVE_MODE):** Default interleave mode that specifies how the Slice Selector and Channel Selector bits are to be determined. Relevant only when HVM mode is disabled and only for system memory addresses that do not fall under the MOT region or the Asymmetric memory region in the System Address Map. Legal encodings are 0x0 0x1 and 0x2. An encoding of 0x3 is treated as if it was 0x2. When both slices and all four PMI channels are enabled: • 0h: Default Slice Selector is Addr[10] and Default Channel Selector is Addr[11] • 1h: Default Slice Selector is Addr[11] and Default Channel Selector is Addr[12] • 2h: Default Slice Selector is Addr[12] and Default Channel Selector is Addr[13] When both slices are enabled but only one channel in each slice enabled: • 0h: Default Slice Selector is Addr[10] • 1h: Default Slice Selector is Addr[11] • 2h: Default Slice Selector is Addr[12] When only SLICE0 is enabled and both channels on that slice are enabled: • 0h: Default Channel Selector is Addr[10] • 1h: Default Channel Selector is Addr[11] • 2h: Default Channel Selector is Addr[12] When SLICE0 and only one channel in that slice is enabled this field is not relevant. B-Unit overrides the setting of the SLICE_HASH_MASK to always include the Slice Selector bit. Similarly, B-Unit overrides the setting of the CH_HASH_MASK to always include the Channel Selector bit.
1	0h RW	**HVM Mode (HVM_MODE):** • 0: HVM mode is disabled. • 1: HVM mode is enabled. When HVM mode is enabled, Slice Hash and Channel Hash is done as follows: Both slices and all four PMI channels enabled: • Slice Hash is Request Physical Addr[29] • Channel Hash is PostRemap Addr[30] Both slices enabled but only one PMI channel in each slice enabled: • Slice Hash is Request Physical Addr[29] Only one SLICE0 enabled but both PMI channels in SLICE0 enabled: • Channel Hash is PostRemap Addr[29] When HVM_MODE is enabled TOLUD must be set at 2GB.
0	0h RW	**Slice 1 Disabled (SLICE_1_DISABLED):** Slice 1 is disabled; no memory address mapped to Slice 1. All request sent to Slice 0.

3.278 Mirror Range Register (A_CR_MIRROR_RANGE_0_0_0_MCHBAR) — Offset 65C8h

Mirror Range: This register defines base and limit of the 8M aligned region in memory that captures the mirror writes. Since b[22:0] are assumed to be 0's the smallest size of the region is 8M

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 65C8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:48	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
47:32	0h RW	**Mirror Limit Address (MIRROR_LIMIT):** Mirror Limit: specifies b38:b23 of HPA indicating the end of mirror packet buffer region
31:16	0h RO	**Reserved (RSVD):** Reserved
15:0	0h RW	**MIRROR_BASE:** Mirror Base: specifies b38:b23 of HPA indicating the start of mirror packet buffer region

3.279 ASYM MEM REGION 0 CONFIGURATION WITH NO INTERLEAVING (A_CR_ASYM_MEM_REGION0_0_0_0_MCHBAR) — Offset 65D0h

Specification of asymmetric memory region 0 (in slice 0) for the configuration with 2 asymmetric memory regions.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 65D0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Enable Asymmetric Region in Slice 0, With No Interleaving (SLICE0_ASYM_ENABLE):** Setting this bit to 0 disables asymmetric memory region 0; setting it to 1 enables the region.
30	0h RW	**Channel Select for ASYM Region Slice 0 (SLICE0_ASYM_CHANNEL_SELECT):** Specifies which Channel in ASYM Slice 0 request is sent
29:19	0h RW	**Limit Address for Asymmetric Memory Region, Slice 0, With No Interleaving (SLICE0_ASYM_LIMIT):** Specifies bits [38:31] of the highest address of asymmetric memory region 0 (in slice 0); all the lower bits of the region's highest address are equal to 1.
18:15	0h RO	**Reserved (RSVD):** Reserved
14:4	0h RW	**Base Address for Asymmetric Memory Region, Slice 0, With No Interleaving (SLICE0_ASYM_BASE):** Specifies bits [38:31] of the base address of asymmetric memory region 0 (in slice 0); all the lower bits of the region's base address are equal to 0.
3:0	0h RO	**Reserved (RSVD):** Reserved

3.280 ASYM MEM REGION 1 CONFIGURATION WITH NO INTERLEAVING (A_CR_ASYM_MEM_REGION1_0_0_0_MCHBAR) — Offset 65D4h

Specification of asymmetric memory region 1 (in slice 1) for the configuration with 2 asymmetric memory regions.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 65D4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Enable Asymmetric Region in Slice 1, With No Interleaving (SLICE1_ASYM_ENABLE):** Setting this bit to 0 disables asymmetric memory region 1; setting it to 1 enables the region.
30	0h RW	**Channel Select for ASYM SLICE 1 (SLICE1_ASYM_CHANNEL_SELECT):** Specifies Channel for ASYM Region Slice 1
29:19	0h RW	**Limit Address for Asymmetric Memory Region, Slice 1, With No Interleaving (SLICE1_ASYM_LIMIT):** Specifies bits [38:31] of the highest address of asymmetric memory region 1 (in slice 1); all the lower bits of the region's highest address are equal to 1.
18:15	0h RO	**Reserved (RSVD):** Reserved
14:4	0h RW	**Limit Address for Asymmetric Memory Region, Slice 1, With No Interleaving (SLICE1_ASYM_BASE):** Specifies bits [38:31] of the base address of asymmetric memory region 1 (in slice 1); all the lower bits of the region's base address are equal to 0.
3:0	0h RO	**Reserved (RSVD):** Reserved

3.281 Two-Way Asymmetric Memory Region Configuration (A_CR_ASYM_2WAY_MEM_REGION_0_0_0_MCHBAR) — Offset 65D8h

Specification of asymmetric memory region 1 (in slice 1) for the configuration with 2 asymmetric memory regions.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 65D8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Enable Two-Way Asymmetric Memory Configuration (ASYM_2WAY_INTERLEAVE_ENABLE):** Setting this bit to 0 disables Interleave Asymmetric memory region 1, setting it to 1 enables the region.
30:28	0h RO	**Reserved (RSVD):** Reserved
27:17	0h RW	**Limit Address for Two-Way Asymmetric Memory Region (ASYM_2WAY_LIMIT):** Specifies bits [38:28] of the highest address of Interleave Asymmetric Region, all the lower bits of the region's highest address are equal to 1.
16:15	0h RO	**Reserved (RSVD):** Reserved
14:4	0h RW	**Base Address for Two-Way Asymmetric Memory Region (ASYM_2WAY_BASE):** Specifies bits [38:28] of the base address of Interleave Asymmetric Region; all the lower bits of the region's base address are equal to 0.

Bit Range	Default & Access	Field Name (ID): Description
3:2	0h RO	**Reserved (RSVD):** Reserved
1:0	0h RW	**Two-Way Asymmetric Interleave Mode (ASYM_2WAY_INTLV_MODE):** Going with 2 bits here. 2'b00 : Asymmetric memory Split between Channel 0 of Slice 0 and Slice 1 2'b01 : Asymmetric memory split between Channel 1 of Slice 0 and Slice 1 2'b10 : Asymmetric memory split between Channel 0 and Channel 1 of Slice 0 2'b11 : Asymmetric memory split between Channel 0 and Channel 1 of Slice 1

3.282 B-Unit Miscellaneous Configuration (B_CR_BMISC_0_0_0_MCHBAR) — Offset 6800h

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6800h	7 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RESERVED_0):** Reserved
3	0h RW	**Send Boot Vector to DRAM (SEND_BOOT_VECTOR_TO_DRAM):** When set, IA accesses to 0xFFFF_0000 to 0XFFFF_FFFF will be sent to memory, regardless of the Host IO Boundary setting in TOLUD.
2	1h RW	**ABSEGINDRAM:** When this bit is set, reads and writes targeting A or Bsegments are routed to DRAM. Asegment corresponds to the memory range 0xA_0000 to 0xA_FFFF. Bsegment corresponds to the memory ranges 0xB_0000 to 0xB_7FFF and 0xB_8000 to 0xB_FFFF.
1	1h RW	**Read FSeg from DRAM (READ_FSEG_FROM_DRAM):** When this bit is set, reads targeting Fsegment are routed to DRAM. Fsegment corresponds to the memory range 0xF_0000 to 0xF_FFFF.
0	1h RW	**Read ESeg from DRAM (READ_ESEG_FROM_DRAM):** When this bit is set, reads targeting Esegment are routed to DRAM. Esegment corresponds to the memory range 0xE_0000 to 0xE_FFFF.

3.283 Security Control Policy (B_CR_SECURITY_CP_0_0_0_MCHBAR) — Offset 6808h

This register controls the access policy to the B-Unit SEC Read Access Control Policy SECURITY_RAC, Write Access Control policy registers SECURITY_WAC, and self-referentially to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6808h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RO	**Security Control Policy (SEC_SAI_POL_63):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
62	0h RO	**Security Control Policy (SEC_SAI_POL_62):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
61	0h RO	**Security Control Policy (SEC_SAI_POL_61):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
60	0h RO	**Security Control Policy (SEC_SAI_POL_60):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
59	0h RO	**Security Control Policy (SEC_SAI_POL_59):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
58	0h RO	**Security Control Policy (SEC_SAI_POL_58):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
57	0h RO	**Security Control Policy (SEC_SAI_POL_57):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
56	0h RO	**Security Control Policy (SEC_SAI_POL_56):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
55	0h RO	**Security Control Policy (SEC_SAI_POL_55):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
54	0h RO	**Security Control Policy (SEC_SAI_POL_54):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
53	0h RO	**Security Control Policy (SEC_SAI_POL_53):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
52	0h RO	**Security Control Policy (SEC_SAI_POL_52):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
51	0h RO	**Security Control Policy (SEC_SAI_POL_51):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
50	0h RO	**Security Control Policy (SEC_SAI_POL_50):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
49	0h RO	**Security Control Policy (SEC_SAI_POL_49):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
48	0h RO	**Security Control Policy (SEC_SAI_POL_48):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
47	0h RO	**Security Control Policy (SEC_SAI_POL_47):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
46	0h RO	**Security Control Policy (SEC_SAI_POL_46):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
45	0h RO	**Security Control Policy (SEC_SAI_POL_45):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
44	0h RO	**Security Control Policy (SEC_SAI_POL_44):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
43	0h RO	**Security Control Policy (SEC_SAI_POL_43):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
42	0h RO	**Security Control Policy (SEC_SAI_POL_42):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
41	0h RO	**Security Control Policy (SEC_SAI_POL_41):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
40	0h RO	**Security Control Policy (SEC_SAI_POL_40):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
39	0h RO	**Security Control Policy (SEC_SAI_POL_39):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
38	0h RO	**Security Control Policy (SEC_SAI_POL_38):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
37	0h RO	**Security Control Policy (SEC_SAI_POL_37):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
36	0h RO	**Security Control Policy (SEC_SAI_POL_36):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
35	0h RO	**Security Control Policy (SEC_SAI_POL_35):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
34	0h RO	**Security Control Policy (SEC_SAI_POL_34):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
33	0h RO	**Security Control Policy (SEC_SAI_POL_33):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
32	0h RO	**Security Control Policy (SEC_SAI_POL_32):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
31	0h RO	**Security Control Policy (SEC_SAI_POL_31):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
30	0h RO	**Security Control Policy (SEC_SAI_POL_30):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
29	0h RO	**Security Control Policy (SEC_SAI_POL_29):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
28	0h RO	**Security Control Policy (SEC_SAI_POL_28):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
27	0h RO	**Security Control Policy (SEC_SAI_POL_27):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
26	0h RO	**Security Control Policy (SEC_SAI_POL_26):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RO	**Security Control Policy (SEC_SAI_POL_25):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
24	0h RO	**Security Control Policy (SEC_SAI_POL_24):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
23	0h RO	**Security Control Policy (SEC_SAI_POL_23):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
22	0h RO	**Security Control Policy (SEC_SAI_POL_22):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
21	0h RO	**Security Control Policy (SEC_SAI_POL_21):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
20	0h RO	**Security Control Policy (SEC_SAI_POL_20):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
19	0h RO	**Security Control Policy (SEC_SAI_POL_19):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
18	0h RO	**Security Control Policy (SEC_SAI_POL_18):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
17	0h RO	**Security Control Policy (SEC_SAI_POL_17):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
16	0h RO	**Security Control Policy (SEC_SAI_POL_16):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
15	0h RO	**Security Control Policy (SEC_SAI_POL_15):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
14	0h RO	**Security Control Policy (SEC_SAI_POL_14):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
13	0h RO	**Security Control Policy (SEC_SAI_POL_13):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
12	0h RO	**Security Control Policy (SEC_SAI_POL_12):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
11	0h RO	**Security Control Policy (SEC_SAI_POL_11):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
10	0h RO	**Security Control Policy (SEC_SAI_POL_10):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
9	0h RO	**Security Control Policy (SEC_SAI_POL_9):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
8	0h RO	**Security Control Policy (SEC_SAI_POL_8):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
7	0h RO	**Security Control Policy (SEC_SAI_POL_7):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RO	**Security Control Policy (SEC_SAI_POL_6):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
5	0h RO	**Security Control Policy (SEC_SAI_POL_5):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
4	0h RO	**Security Control Policy (SEC_SAI_POL_4):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
3	0h RO	**Security Control Policy (SEC_SAI_POL_3):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
2	0h RO	**Security Control Policy (SEC_SAI_POL_2):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
1	0h RO	**Security Control Policy (SEC_SAI_POL_1):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.
0	0h RO	**Security Control Policy (SEC_SAI_POL_0):** Bit vector used to determine which agents are allowed access to the SECURITY_RAC, SECURITY_WAC and SECURITY_CP registers, based on the value from each agent's 6bit SAI field.

3.284 Security Group Read Access Policy (B_CR_SECURITY_RAC_0_0_0_MCHBAR) — Offset 6810h

This register configures the Read Access Policy for the B-Unit Security Group registers. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6810h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_63):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
62	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_62):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
61	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_61):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
60	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_60):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
59	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_59):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
58	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_58):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
57	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_57):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
56	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_56):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
55	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_55):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
54	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_54):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
53	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_53):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
52	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_52):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
51	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_51):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
50	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_50):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
49	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_49):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
48	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_48):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
47	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_47):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
46	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_46):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
45	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_45):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
44	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_44):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
43	1h RO	**Security SAI Read Access Policy (SEC_SAI_POL_43):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
42	1h RO	**Security SAI Read Access Policy (SEC_SAI_POL_42):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
41	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_41):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
40	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_40):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
39	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_39):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
38	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_38):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
37	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_37):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
36	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_36):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
35	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_35):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
34	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_34):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
33	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_33):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
32	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_32):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
31	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_31):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
30	1h RO	**Security SAI Read Access Policy (SEC_SAI_POL_30):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
29	1h RO	**Security SAI Read Access Policy (SEC_SAI_POL_29):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
28	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_28):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
27	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_27):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
26	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_26):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
25	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_25):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
24	1h RO	**Security SAI Read Access Policy (SEC_SAI_POL_24):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
23	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_23):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
22	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_22):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
21	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_21):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_20):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
19	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_19):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
18	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_18):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
17	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_17):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
16	1h RO	**Security SAI Read Access Policy (SEC_SAI_POL_16):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
15	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_15):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
14	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_14):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
13	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_13):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
12	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_12):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
11	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_11):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
10	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_10):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
9	1h RO	**Security SAI Read Access Policy (SEC_SAI_POL_9):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
8	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_8):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
7	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_7):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
6	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_6):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
5	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_5):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
4	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_4):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
3	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_3):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
2	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_2):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
1	1h RO	**Security SAI Read Access Policy (SEC_SAI_POL_1):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.
0	0h RO	**Security SAI Read Access Policy (SEC_SAI_POL_0):** Bit vector used to determine which agents are allowed read access to the B-Unit Security Group Configuration registers based on each agent's 6bit encoded SAI value.

3.285 Security Group Write Access Policy (B_CR_SECURITY_WAC_0_0_0_MCHBAR) — Offset 6818h

This register configures the Write Access Policy for the B-Unit Security Group registers. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6818h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_63):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
62	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_62):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
61	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_61):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
60	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_60):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
59	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_59):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
58	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_58):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
57	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_57):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
56	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_56):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
55	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_55):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
54	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_54):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
53	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_53):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
52	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_52):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
51	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_51):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
50	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_50):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
49	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_49):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
48	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_48):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
47	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_47):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
46	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_46):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
45	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_45):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
44	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_44):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
43	1h RO	**Security SAI Write Access Policy (SEC_SAI_POL_43):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
42	1h RO	**Security SAI Write Access Policy (SEC_SAI_POL_42):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
41	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_41):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
40	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_40):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
39	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_39):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
38	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_38):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
37	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_37):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
36	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_36):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
35	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_35):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
34	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_34):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
33	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_33):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
32	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_32):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
31	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_31):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
30	1h RO	**Security SAI Write Access Policy (SEC_SAI_POL_30):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
29	1h RO	**Security SAI Write Access Policy (SEC_SAI_POL_29):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
28	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_28):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
27	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_27):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
26	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_26):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
25	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_25):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
24	1h RO	**Security SAI Write Access Policy (SEC_SAI_POL_24):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
23	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_23):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
22	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_22):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
21	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_21):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
20	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_20):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
19	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_19):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
18	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_18):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
17	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_17):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
16	1h RO	**Security SAI Write Access Policy (SEC_SAI_POL_16):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_15):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
14	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_14):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
13	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_13):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
12	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_12):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
11	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_11):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
10	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_10):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
9	1h RO	**Security SAI Write Access Policy (SEC_SAI_POL_9):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
8	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_8):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
7	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_7):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
6	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_6):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
5	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_5):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
4	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_4):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
3	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_3):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
2	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_2):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
1	1h RO	**Security SAI Write Access Policy (SEC_SAI_POL_1):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.
0	0h RO	**Security SAI Write Access Policy (SEC_SAI_POL_0):** Bit vector used to determine which agents are allowed write access to the B-Unit Security Group Configuration registers, based on each agent's 6bit encoded SAI value.

3.286 Slice 0 Memory Access Count (B_CR_MEM_ACCESS_COUNT_SLICE0) — Offset 6868h

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6868h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Memory Access Count (MEM_ACCESS_COUNT):** Counts the number of PMI transactions that the B-Unit has sent to any PMI channel. Counts both reads and writes. Counter is not saturating and will roll over to zero. It is up to the consumer of the counter to handle roll over cases.

3.287 Slice 1 Memory Access Count (B_CR_MEM_ACCESS_COUNT_SLICE1) — Offset 686Ch

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 686Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Memory Access Count (MEM_ACCESS_COUNT):** Counts the number of PMI transactions that the B-Unit has sent to any PMI channel. Counts both reads and writes. Counter is not saturating and will roll over to zero. It is up to the consumer of the counter to handle roll over cases.

3.288 IMR0 Base (B_CR_BIMR0BASE_0_0_0_MCHBAR) — Offset 6870h

This register, along with IMR0MASK, IMR0RAC, and IMR0WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR0RAC and IMR0WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6870h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.

Bit Range	Default & Access	Field Name (ID): Description
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR0 Base (IMR0_BASE):** Specifies bits 38:10 of the start address of IMR0 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR0MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR0 defined region.

3.289 IMR0 Mask (B_CR_BIMR0MASK_0_0_0_MCHBAR) — Offset 6874h

This register, along with IMR0BASE, IMR0RAC, and IMR0WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR0RAC and IMR0WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6874h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved
28:0	0h RW	**Mask 0 IMR0 Mask (IMR0_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR0BASE[28:0] value. A match indicates that the incoming address falls within the IMR0 region.

3.290 IMR0 Control Policy (B_CR_BIMR0CP_0_0_0_MCHBAR) — Offset 6878h

This register controls the access policy to the Read Access Policy BIMR0RAC, Write Access Policy BIMR0WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6878h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
60	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
59	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
58	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
55	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
52	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
50	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
47	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
44	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR0 Control Policy (IMR0_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR0 Control Policy (IMR0_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
41	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
40	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
39	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
36	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
33	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
31	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR0 Control Policy (IMR0_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR0 Control Policy (IMR0_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
28	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
25	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR0 Control Policy (IMR0_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
22	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
21	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
20	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
17	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR0 Control Policy (IMR0_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
14	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
12	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
9	1h RW	**IMR0 Control Policy (IMR0_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
6	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
3	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
2	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
1	1h RW	**IMR0 Control Policy (IMR0_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR0 Control Policy (IMR0_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR0RAC, BIMR0WAC and BIMR0CP registers based on the value from each agent's 6bit SAI field.

3.291 IMR0 Read Access Policy (B_CR_BIMR0RAC_0_0_0_MCHBAR) — Offset 6880h

This register, along with IMR0BASE, IMR0MASK and IMR0WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR0. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6880h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR0 Read Access Policy 63 (IMR0_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR0 Read Access Policy 62 (IMR0_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
61	0h RO	**IMR0 Read Access Policy 61 (IMR0_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR0 Read Access Policy 60 (IMR0_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR0 Read Access Policy 59 (IMR0_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR0 Read Access Policy 58 (IMR0_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR0 Read Access Policy 57 (IMR0_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR0 Read Access Policy 56 (IMR0_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR0 Read Access Policy 55 (IMR0_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR0 Read Access Policy 54 (IMR0_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR0 Read Access Policy 53 (IMR0_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR0 Read Access Policy 52 (IMR0_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR0 Read Access Policy 51 (IMR0_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR0 Read Access Policy 50 (IMR0_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR0 Read Access Policy 49 (IMR0_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR0 Read Access Policy 48 (IMR0_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR0 Read Access Policy 47 (IMR0_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR0 Read Access Policy 46 (IMR0_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR0 Read Access Policy 45 (IMR0_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR0 Read Access Policy 44 (IMR0_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR0 Read Access Policy 43 (IMR0_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
42	0h RW	**IMR0 Read Access Policy 42 (IMR0_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR0 Read Access Policy 41 (IMR0_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR0 Read Access Policy 40 (IMR0_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR0 Read Access Policy 39 (IMR0_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR0 Read Access Policy 38 (IMR0_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR0 Read Access Policy 37 (IMR0_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR0 Read Access Policy 36 (IMR0_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR0 Read Access Policy 35 (IMR0_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR0 Read Access Policy 34 (IMR0_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR0 Read Access Policy 33 (IMR0_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR0 Read Access Policy 32 (IMR0_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR0 Read Access Policy 31 (IMR0_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR0 Read Access Policy 30 (IMR0_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR0 Read Access Policy 29 (IMR0_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR0 Read Access Policy 28 (IMR0_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR0 Read Access Policy 27 (IMR0_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR0 Read Access Policy 26 (IMR0_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR0 Read Access Policy 25 (IMR0_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR0 Read Access Policy 24 (IMR0_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RO	**IMR0 Read Access Policy 23 (IMR0_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR0 Read Access Policy 22 (IMR0_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR0 Read Access Policy 21 (IMR0_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR0 Read Access Policy 20 (IMR0_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR0 Read Access Policy 19 (IMR0_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR0 Read Access Policy 18 (IMR0_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR0 Read Access Policy 17 (IMR0_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR0 Read Access Policy 16 (IMR0_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR0 Read Access Policy 15 (IMR0_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR0 Read Access Policy 14 (IMR0_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR0 Read Access Policy 13 (IMR0_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR0 Read Access Policy 12 (IMR0_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR0 Read Access Policy 11 (IMR0_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR0 Read Access Policy 10 (IMR0_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR0 Read Access Policy 9 (IMR0_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR0 Read Access Policy 8 (IMR0_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR0 Read Access Policy 7 (IMR0_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR0 Read Access Policy 6 (IMR0_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR0 Read Access Policy 5 (IMR0_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW	**IMR0 Read Access Policy 4 (IMR0_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR0 Read Access Policy 3 (IMR0_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR0 Read Access Policy 2 (IMR0_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR0 Read Access Policy 1 (IMR0_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR0 Read Access Policy 0 (IMR0_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR0 region, based on each agent's 6bit encoded SAI value.

3.292 IMR0 Write Access Policy (B_CR_BIMR0WAC_0_0_0_MCHBAR) — Offset 6888h

This register, along with IMR0BASE, IMR0MASK and IMR0RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR0. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6888h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR0 Write Access Policy 63 (IMR0_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR0 Write Access Policy 62 (IMR0_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR0 Write Access Policy 61 (IMR0_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR0 Write Access Policy 60 (IMR0_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR0 Write Access Policy 59 (IMR0_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR0 Write Access Policy 58 (IMR0_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR0 Write Access Policy 57 (IMR0_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
56	0h RW	**IMR0 Write Access Policy 56 (IMR0_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR0 Write Access Policy 55 (IMR0_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR0 Write Access Policy 54 (IMR0_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR0 Write Access Policy 53 (IMR0_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR0 Write Access Policy 52 (IMR0_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR0 Write Access Policy 51 (IMR0_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR0 Write Access Policy 50 (IMR0_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR0 Write Access Policy 49 (IMR0_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR0 Write Access Policy 48 (IMR0_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR0 Write Access Policy 47 (IMR0_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR0 Write Access Policy 46 (IMR0_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR0 Write Access Policy 45 (IMR0_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR0 Write Access Policy 44 (IMR0_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR0 Write Access Policy 43 (IMR0_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR0 Write Access Policy 42 (IMR0_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR0 Write Access Policy 41 (IMR0_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR0 Write Access Policy 40 (IMR0_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR0 Write Access Policy 39 (IMR0_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR0 Write Access Policy 38 (IMR0_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
37	0h RO	**IMR0 Write Access Policy 37 (IMR0_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR0 Write Access Policy 36 (IMR0_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR0 Write Access Policy 35 (IMR0_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR0 Write Access Policy 34 (IMR0_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR0 Write Access Policy 33 (IMR0_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR0 Write Access Policy 32 (IMR0_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR0 Write Access Policy 31 (IMR0_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR0 Write Access Policy 30 (IMR0_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR0 Write Access Policy 29 (IMR0_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR0 Write Access Policy 28 (IMR0_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR0 Write Access Policy 27 (IMR0_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR0 Write Access Policy 26 (IMR0_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR0 Write Access Policy 25 (IMR0_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR0 Write Access Policy 24 (IMR0_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR0 Write Access Policy 23 (IMR0_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR0 Write Access Policy 22 (IMR0_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR0 Write Access Policy 21 (IMR0_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR0 Write Access Policy 20 (IMR0_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR0 Write Access Policy 19 (IMR0_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**IMR0 Write Access Policy 18 (IMR0_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR0 Write Access Policy 17 (IMR0_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR0 Write Access Policy 16 (IMR0_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR0 Write Access Policy 15 (IMR0_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR0 Write Access Policy 14 (IMR0_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR0 Write Access Policy 13 (IMR0_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR0 Write Access Policy 12 (IMR0_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR0 Write Access Policy 11 (IMR0_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR0 Write Access Policy 10 (IMR0_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR0 Write Access Policy 9 (IMR0_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR0 Write Access Policy 8 (IMR0_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR0 Write Access Policy 7 (IMR0_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR0 Write Access Policy 6 (IMR0_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR0 Write Access Policy 5 (IMR0_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR0 Write Access Policy 4 (IMR0_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR0 Write Access Policy 3 (IMR0_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR0 Write Access Policy 2 (IMR0_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR0 Write Access Policy 1 (IMR0_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR0 Write Access Policy 0 (IMR0_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR0 region, based on each agent's 6bit encoded SAI value.

3.293 IMR1 Base (B_CR_BIMR1BASE_0_0_0_MCHBAR) — Offset 6890h

This register, along with IMR1MASK, IMR1RAC, and IMR1WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR1RAC and IMR1WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6890h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR1 Base (IMR1_BASE):** Specifies bits 38:10 of the start address of IMR1 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR1MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR1 defined region.

3.294 IMR1 Mask (B_CR_BIMR1MASK_0_0_0_MCHBAR) — Offset 6894h

This register, along with IMR1BASE, IMR1RAC, and IMR1WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR1RAC and IMR1WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6894h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.

Bit Range	Default & Access	Field Name (ID): Description
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved
28:0	0h RW	**Mask 0 IMR1 Mask (IMR1_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR1BASE[28:0] value. A match indicates that the incoming address falls within the IMR1 region.

3.295 IMR1 Control Policy (B_CR_BIMR1CP_0_0_0_MCHBAR) — Offset 6898h

This register controls the access policy to the Read Access Policy BIMR1RAC, Write Access Policy BIMR1WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6898h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
60	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
59	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
58	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
55	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
52	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
50	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
47	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
44	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR1 Control Policy (IMR1_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR1 Control Policy (IMR1_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
41	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
40	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
39	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
36	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
33	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
31	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR1 Control Policy (IMR1_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR1 Control Policy (IMR1_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
28	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
25	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR1 Control Policy (IMR1_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
22	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
21	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
20	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR1 Control Policy (IMR1_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
14	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
12	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
9	1h RW	**IMR1 Control Policy (IMR1_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
6	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
3	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
2	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
1	1h RW	**IMR1 Control Policy (IMR1_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR1 Control Policy (IMR1_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR1RAC, BIMR1WAC and BIMR1CP registers based on the value from each agent's 6bit SAI field.

3.296 IMR1 Read Access Policy (B_CR_BIMR1RAC_0_0_0_MCHBAR) — Offset 68A0h

This register, along with IMR1BASE, IMR1MASK and IMR1WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR1. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 68A0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR1 Read Access Policy 63 (IMR1_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR1 Read Access Policy 62 (IMR1_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR1 Read Access Policy 61 (IMR1_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR1 Read Access Policy 60 (IMR1_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR1 Read Access Policy 59 (IMR1_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR1 Read Access Policy 58 (IMR1_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR1 Read Access Policy 57 (IMR1_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR1 Read Access Policy 56 (IMR1_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR1 Read Access Policy 55 (IMR1_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR1 Read Access Policy 54 (IMR1_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR1 Read Access Policy 53 (IMR1_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR1 Read Access Policy 52 (IMR1_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR1 Read Access Policy 51 (IMR1_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
50	0h RW	**IMR1 Read Access Policy 50 (IMR1_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR1 Read Access Policy 49 (IMR1_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR1 Read Access Policy 48 (IMR1_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR1 Read Access Policy 47 (IMR1_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR1 Read Access Policy 46 (IMR1_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR1 Read Access Policy 45 (IMR1_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR1 Read Access Policy 44 (IMR1_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR1 Read Access Policy 43 (IMR1_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR1 Read Access Policy 42 (IMR1_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR1 Read Access Policy 41 (IMR1_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR1 Read Access Policy 40 (IMR1_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR1 Read Access Policy 39 (IMR1_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR1 Read Access Policy 38 (IMR1_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR1 Read Access Policy 37 (IMR1_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR1 Read Access Policy 36 (IMR1_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR1 Read Access Policy 35 (IMR1_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR1 Read Access Policy 34 (IMR1_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR1 Read Access Policy 33 (IMR1_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR1 Read Access Policy 32 (IMR1_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**IMR1 Read Access Policy 31 (IMR1_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR1 Read Access Policy 30 (IMR1_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR1 Read Access Policy 29 (IMR1_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR1 Read Access Policy 28 (IMR1_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR1 Read Access Policy 27 (IMR1_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR1 Read Access Policy 26 (IMR1_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR1 Read Access Policy 25 (IMR1_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR1 Read Access Policy 24 (IMR1_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR1 Read Access Policy 23 (IMR1_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR1 Read Access Policy 22 (IMR1_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR1 Read Access Policy 21 (IMR1_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR1 Read Access Policy 20 (IMR1_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR1 Read Access Policy 19 (IMR1_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR1 Read Access Policy 18 (IMR1_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR1 Read Access Policy 17 (IMR1_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR1 Read Access Policy 16 (IMR1_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR1 Read Access Policy 15 (IMR1_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR1 Read Access Policy 14 (IMR1_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR1 Read Access Policy 13 (IMR1_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**IMR1 Read Access Policy 12 (IMR1_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR1 Read Access Policy 11 (IMR1_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR1 Read Access Policy 10 (IMR1_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR1 Read Access Policy 9 (IMR1_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR1 Read Access Policy 8 (IMR1_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR1 Read Access Policy 7 (IMR1_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR1 Read Access Policy 6 (IMR1_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR1 Read Access Policy 5 (IMR1_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR1 Read Access Policy 4 (IMR1_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR1 Read Access Policy 3 (IMR1_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR1 Read Access Policy 2 (IMR1_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR1 Read Access Policy 1 (IMR1_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR1 Read Access Policy 0 (IMR1_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR1 region, based on each agent's 6bit encoded SAI value.

3.297 IMR1 Write Access Policy (B_CR_BIMR1WAC_0_0_0_MCHBAR) — Offset 68A8h

This register, along with IMR1BASE, IMR1MASK and IMR1RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR1. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 68A8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR1 Write Access Policy 63 (IMR1_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR1 Write Access Policy 62 (IMR1_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR1 Write Access Policy 61 (IMR1_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR1 Write Access Policy 60 (IMR1_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR1 Write Access Policy 59 (IMR1_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR1 Write Access Policy 58 (IMR1_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR1 Write Access Policy 57 (IMR1_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR1 Write Access Policy 56 (IMR1_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR1 Write Access Policy 55 (IMR1_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR1 Write Access Policy 54 (IMR1_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR1 Write Access Policy 53 (IMR1_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR1 Write Access Policy 52 (IMR1_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR1 Write Access Policy 51 (IMR1_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR1 Write Access Policy 50 (IMR1_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR1 Write Access Policy 49 (IMR1_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR1 Write Access Policy 48 (IMR1_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR1 Write Access Policy 47 (IMR1_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR1 Write Access Policy 46 (IMR1_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR1 Write Access Policy 45 (IMR1_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
44	0h RW	**IMR1 Write Access Policy 44 (IMR1_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR1 Write Access Policy 43 (IMR1_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR1 Write Access Policy 42 (IMR1_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR1 Write Access Policy 41 (IMR1_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR1 Write Access Policy 40 (IMR1_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR1 Write Access Policy 39 (IMR1_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR1 Write Access Policy 38 (IMR1_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR1 Write Access Policy 37 (IMR1_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR1 Write Access Policy 36 (IMR1_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR1 Write Access Policy 35 (IMR1_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR1 Write Access Policy 34 (IMR1_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR1 Write Access Policy 33 (IMR1_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR1 Write Access Policy 32 (IMR1_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR1 Write Access Policy 31 (IMR1_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR1 Write Access Policy 30 (IMR1_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR1 Write Access Policy 29 (IMR1_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR1 Write Access Policy 28 (IMR1_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR1 Write Access Policy 27 (IMR1_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR1 Write Access Policy 26 (IMR1_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**IMR1 Write Access Policy 25 (IMR1_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR1 Write Access Policy 24 (IMR1_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR1 Write Access Policy 23 (IMR1_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR1 Write Access Policy 22 (IMR1_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR1 Write Access Policy 21 (IMR1_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR1 Write Access Policy 20 (IMR1_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR1 Write Access Policy 19 (IMR1_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR1 Write Access Policy 18 (IMR1_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR1 Write Access Policy 17 (IMR1_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR1 Write Access Policy 16 (IMR1_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR1 Write Access Policy 15 (IMR1_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR1 Write Access Policy 14 (IMR1_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR1 Write Access Policy 13 (IMR1_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR1 Write Access Policy 12 (IMR1_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR1 Write Access Policy 11 (IMR1_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR1 Write Access Policy 10 (IMR1_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR1 Write Access Policy 9 (IMR1_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR1 Write Access Policy 8 (IMR1_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR1 Write Access Policy 7 (IMR1_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**IMR1 Write Access Policy 6 (IMR1_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR1 Write Access Policy 5 (IMR1_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR1 Write Access Policy 4 (IMR1_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR1 Write Access Policy 3 (IMR1_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR1 Write Access Policy 2 (IMR1_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR1 Write Access Policy 1 (IMR1_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR1 Write Access Policy 0 (IMR1_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR1 region, based on each agent's 6bit encoded SAI value.

3.298 IMR2 Base (B_CR_BIMR2BASE_0_0_0_MCHBAR) — Offset 68B0h

This register, along with IMR2MASK, IMR2RAC, and IMR2WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR2RAC and IMR2WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 68B0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR2 Base (IMR2_BASE):** Specifies bits 38:10 of the start address of IMR2 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR2MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR2 defined region.

3.299 IMR2 Mask (B_CR_BIMR2MASK_0_0_0_MCHBAR) — Offset 68B4h

This register, along with IMR2BASE, IMR2RAC, and IMR2WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR2RAC and IMR2WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 68B4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved
28:0	0h RW	**Mask 0 IMR2 Mask (IMR2_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR2BASE[28:0] value. A match indicates that the incoming address falls within the IMR2 region.

3.300 IMR2 Control Policy (B_CR_BIMR2CP_0_0_0_MCHBAR) — Offset 68B8h

This register controls the access policy to the Read Access Policy BIMR2RAC, Write Access Policy BIMR2WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 68B8h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
60	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
59	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
58	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
55	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
52	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
50	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
47	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
44	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR2 Control Policy (IMR2_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR2 Control Policy (IMR2_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
41	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
40	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
39	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
36	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
33	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
31	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR2 Control Policy (IMR2_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR2 Control Policy (IMR2_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
28	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR2 Control Policy (IMR2_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
22	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
21	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
20	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
17	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR2 Control Policy (IMR2_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
14	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
12	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
9	1h RW	**IMR2 Control Policy (IMR2_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
3	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
2	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
1	1h RW	**IMR2 Control Policy (IMR2_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR2 Control Policy (IMR2_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR2RAC, BIMR2WAC and BIMR2CP registers based on the value from each agent's 6bit SAI field.

3.301 IMR2 Read Access Policy (B_CR_BIMR2RAC_0_0_0_MCHBAR) — Offset 68C0h

This register, along with IMR2BASE, IMR2MASK and IMR2WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR2. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 68C0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR2 Read Access Policy 63 (IMR2_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR2 Read Access Policy 62 (IMR2_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR2 Read Access Policy 61 (IMR2_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR2 Read Access Policy 60 (IMR2_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR2 Read Access Policy 59 (IMR2_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
58	0h RO	**IMR2 Read Access Policy 58 (IMR2_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR2 Read Access Policy 57 (IMR2_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR2 Read Access Policy 56 (IMR2_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR2 Read Access Policy 55 (IMR2_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR2 Read Access Policy 54 (IMR2_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR2 Read Access Policy 53 (IMR2_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR2 Read Access Policy 52 (IMR2_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR2 Read Access Policy 51 (IMR2_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR2 Read Access Policy 50 (IMR2_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR2 Read Access Policy 49 (IMR2_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR2 Read Access Policy 48 (IMR2_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR2 Read Access Policy 47 (IMR2_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR2 Read Access Policy 46 (IMR2_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR2 Read Access Policy 45 (IMR2_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR2 Read Access Policy 44 (IMR2_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR2 Read Access Policy 43 (IMR2_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR2 Read Access Policy 42 (IMR2_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR2 Read Access Policy 41 (IMR2_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR2 Read Access Policy 40 (IMR2_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
39	0h RO	**IMR2 Read Access Policy 39 (IMR2_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR2 Read Access Policy 38 (IMR2_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR2 Read Access Policy 37 (IMR2_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR2 Read Access Policy 36 (IMR2_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR2 Read Access Policy 35 (IMR2_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR2 Read Access Policy 34 (IMR2_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR2 Read Access Policy 33 (IMR2_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR2 Read Access Policy 32 (IMR2_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR2 Read Access Policy 31 (IMR2_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR2 Read Access Policy 30 (IMR2_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR2 Read Access Policy 29 (IMR2_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR2 Read Access Policy 28 (IMR2_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR2 Read Access Policy 27 (IMR2_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR2 Read Access Policy 26 (IMR2_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR2 Read Access Policy 25 (IMR2_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR2 Read Access Policy 24 (IMR2_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR2 Read Access Policy 23 (IMR2_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR2 Read Access Policy 22 (IMR2_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR2 Read Access Policy 21 (IMR2_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**IMR2 Read Access Policy 20 (IMR2_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR2 Read Access Policy 19 (IMR2_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR2 Read Access Policy 18 (IMR2_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR2 Read Access Policy 17 (IMR2_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR2 Read Access Policy 16 (IMR2_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR2 Read Access Policy 15 (IMR2_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR2 Read Access Policy 14 (IMR2_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR2 Read Access Policy 13 (IMR2_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR2 Read Access Policy 12 (IMR2_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR2 Read Access Policy 11 (IMR2_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR2 Read Access Policy 10 (IMR2_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR2 Read Access Policy 9 (IMR2_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR2 Read Access Policy 8 (IMR2_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR2 Read Access Policy 7 (IMR2_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR2 Read Access Policy 6 (IMR2_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR2 Read Access Policy 5 (IMR2_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR2 Read Access Policy 4 (IMR2_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR2 Read Access Policy 3 (IMR2_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR2 Read Access Policy 2 (IMR2_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**IMR2 Read Access Policy 1 (IMR2_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR2 Read Access Policy 0 (IMR2_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR2 region, based on each agent's 6bit encoded SAI value.

3.302 IMR2 Write Access Policy (B_CR_BIMR2WAC_0_0_0_MCHBAR) — Offset 68C8h

This register, along with IMR2BASE, IMR2MASK and IMR2RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR2. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 68C8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR2 Write Access Policy 63 (IMR2_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR2 Write Access Policy 62 (IMR2_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR2 Write Access Policy 61 (IMR2_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR2 Write Access Policy 60 (IMR2_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR2 Write Access Policy 59 (IMR2_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR2 Write Access Policy 58 (IMR2_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR2 Write Access Policy 57 (IMR2_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR2 Write Access Policy 56 (IMR2_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR2 Write Access Policy 55 (IMR2_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR2 Write Access Policy 54 (IMR2_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
53	0h RO	**IMR2 Write Access Policy 53 (IMR2_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR2 Write Access Policy 52 (IMR2_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR2 Write Access Policy 51 (IMR2_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR2 Write Access Policy 50 (IMR2_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR2 Write Access Policy 49 (IMR2_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR2 Write Access Policy 48 (IMR2_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR2 Write Access Policy 47 (IMR2_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR2 Write Access Policy 46 (IMR2_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR2 Write Access Policy 45 (IMR2_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR2 Write Access Policy 44 (IMR2_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR2 Write Access Policy 43 (IMR2_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR2 Write Access Policy 42 (IMR2_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR2 Write Access Policy 41 (IMR2_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR2 Write Access Policy 40 (IMR2_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR2 Write Access Policy 39 (IMR2_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR2 Write Access Policy 38 (IMR2_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR2 Write Access Policy 37 (IMR2_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR2 Write Access Policy 36 (IMR2_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR2 Write Access Policy 35 (IMR2_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
34	0h RW	**IMR2 Write Access Policy 34 (IMR2_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR2 Write Access Policy 33 (IMR2_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR2 Write Access Policy 32 (IMR2_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR2 Write Access Policy 31 (IMR2_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR2 Write Access Policy 30 (IMR2_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR2 Write Access Policy 29 (IMR2_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR2 Write Access Policy 28 (IMR2_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR2 Write Access Policy 27 (IMR2_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR2 Write Access Policy 26 (IMR2_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR2 Write Access Policy 25 (IMR2_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR2 Write Access Policy 24 (IMR2_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR2 Write Access Policy 23 (IMR2_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR2 Write Access Policy 22 (IMR2_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR2 Write Access Policy 21 (IMR2_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR2 Write Access Policy 20 (IMR2_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR2 Write Access Policy 19 (IMR2_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR2 Write Access Policy 18 (IMR2_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR2 Write Access Policy 17 (IMR2_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR2 Write Access Policy 16 (IMR2_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**IMR2 Write Access Policy 15 (IMR2_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR2 Write Access Policy 14 (IMR2_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR2 Write Access Policy 13 (IMR2_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR2 Write Access Policy 12 (IMR2_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR2 Write Access Policy 11 (IMR2_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR2 Write Access Policy 10 (IMR2_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR2 Write Access Policy 9 (IMR2_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR2 Write Access Policy 8 (IMR2_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR2 Write Access Policy 7 (IMR2_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR2 Write Access Policy 6 (IMR2_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR2 Write Access Policy 5 (IMR2_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR2 Write Access Policy 4 (IMR2_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR2 Write Access Policy 3 (IMR2_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR2 Write Access Policy 2 (IMR2_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR2 Write Access Policy 1 (IMR2_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR2 Write Access Policy 0 (IMR2_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR2 region, based on each agent's 6bit encoded SAI value.

3.303 IMR3 Base (B_CR_BIMR3BASE_0_0_0_MCHBAR) — Offset 68D0h

This register, along with IMR3MASK, IMR3RAC, and IMR3WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or

from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR3RAC and IMR3WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 68D0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR3 Base (IMR3_BASE):** Specifies bits 38:10 of the start address of IMR3 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR3MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR3 defined region.

3.304 IMR3 Mask (B_CR_BIMR3MASK_0_0_0_MCHBAR) — Offset 68D4h

This register, along with IMR3BASE, IMR3RAC, and IMR3WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR3RAC and IMR3WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 68D4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
28:0	0h RW	**Mask 0 IMR3 Mask (IMR3_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR3BASE[28:0] value. A match indicates that the incoming address falls within the IMR3 region.

3.305 IMR3 Control Policy (B_CR_BIMR3CP_0_0_0_MCHBAR) — Offset 68D8h

This register controls the access policy to the Read Access Policy BIMR3RAC, Write Access Policy BIMR3WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 68D8h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
60	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
59	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
58	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
55	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
52	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
50	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
47	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
44	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR3 Control Policy (IMR3_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR3 Control Policy (IMR3_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
41	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
40	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
39	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
36	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
33	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
31	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR3 Control Policy (IMR3_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR3 Control Policy (IMR3_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
28	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
25	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR3 Control Policy (IMR3_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
22	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
21	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
20	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
17	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR3 Control Policy (IMR3_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
12	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
9	1h RW	**IMR3 Control Policy (IMR3_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
6	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
3	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
2	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
1	1h RW	**IMR3 Control Policy (IMR3_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR3 Control Policy (IMR3_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR3RAC, BIMR3WAC and BIMR3CP registers based on the value from each agent's 6bit SAI field.

3.306 IMR3 Read Access Policy (B_CR_BIMR3RAC_0_0_0_MCHBAR) — Offset 68E0h

This register, along with IMR3BASE, IMR3MASK and IMR3WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR3. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 68E0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR3 Read Access Policy 63 (IMR3_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR3 Read Access Policy 62 (IMR3_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR3 Read Access Policy 61 (IMR3_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR3 Read Access Policy 60 (IMR3_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR3 Read Access Policy 59 (IMR3_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR3 Read Access Policy 58 (IMR3_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR3 Read Access Policy 57 (IMR3_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR3 Read Access Policy 56 (IMR3_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR3 Read Access Policy 55 (IMR3_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR3 Read Access Policy 54 (IMR3_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR3 Read Access Policy 53 (IMR3_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR3 Read Access Policy 52 (IMR3_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR3 Read Access Policy 51 (IMR3_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR3 Read Access Policy 50 (IMR3_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR3 Read Access Policy 49 (IMR3_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR3 Read Access Policy 48 (IMR3_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR3 Read Access Policy 47 (IMR3_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
46	0h RO	**IMR3 Read Access Policy 46 (IMR3_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR3 Read Access Policy 45 (IMR3_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR3 Read Access Policy 44 (IMR3_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR3 Read Access Policy 43 (IMR3_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR3 Read Access Policy 42 (IMR3_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR3 Read Access Policy 41 (IMR3_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR3 Read Access Policy 40 (IMR3_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR3 Read Access Policy 39 (IMR3_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR3 Read Access Policy 38 (IMR3_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR3 Read Access Policy 37 (IMR3_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR3 Read Access Policy 36 (IMR3_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR3 Read Access Policy 35 (IMR3_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR3 Read Access Policy 34 (IMR3_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR3 Read Access Policy 33 (IMR3_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR3 Read Access Policy 32 (IMR3_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR3 Read Access Policy 31 (IMR3_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR3 Read Access Policy 30 (IMR3_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR3 Read Access Policy 29 (IMR3_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR3 Read Access Policy 28 (IMR3_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**IMR3 Read Access Policy 27 (IMR3_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR3 Read Access Policy 26 (IMR3_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR3 Read Access Policy 25 (IMR3_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR3 Read Access Policy 24 (IMR3_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR3 Read Access Policy 23 (IMR3_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR3 Read Access Policy 22 (IMR3_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR3 Read Access Policy 21 (IMR3_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR3 Read Access Policy 20 (IMR3_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR3 Read Access Policy 19 (IMR3_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR3 Read Access Policy 18 (IMR3_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR3 Read Access Policy 17 (IMR3_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR3 Read Access Policy 16 (IMR3_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR3 Read Access Policy 15 (IMR3_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR3 Read Access Policy 14 (IMR3_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR3 Read Access Policy 13 (IMR3_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR3 Read Access Policy 12 (IMR3_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR3 Read Access Policy 11 (IMR3_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR3 Read Access Policy 10 (IMR3_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR3 Read Access Policy 9 (IMR3_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW	**IMR3 Read Access Policy 8 (IMR3_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR3 Read Access Policy 7 (IMR3_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR3 Read Access Policy 6 (IMR3_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR3 Read Access Policy 5 (IMR3_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR3 Read Access Policy 4 (IMR3_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR3 Read Access Policy 3 (IMR3_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR3 Read Access Policy 2 (IMR3_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR3 Read Access Policy 1 (IMR3_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR3 Read Access Policy 0 (IMR3_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR3 region, based on each agent's 6bit encoded SAI value.

3.307 IMR3 Write Access Policy (B_CR_BIMR3WAC_0_0_0_MCHBAR) — Offset 68E8h

This register, along with IMR3BASE, IMR3MASK and IMR3RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR3. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 68E8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR3 Write Access Policy 63 (IMR3_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR3 Write Access Policy 62 (IMR3_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR3 Write Access Policy 61 (IMR3_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
60	0h RO	**IMR3 Write Access Policy 60 (IMR3_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR3 Write Access Policy 59 (IMR3_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR3 Write Access Policy 58 (IMR3_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR3 Write Access Policy 57 (IMR3_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR3 Write Access Policy 56 (IMR3_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR3 Write Access Policy 55 (IMR3_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR3 Write Access Policy 54 (IMR3_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR3 Write Access Policy 53 (IMR3_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR3 Write Access Policy 52 (IMR3_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR3 Write Access Policy 51 (IMR3_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR3 Write Access Policy 50 (IMR3_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR3 Write Access Policy 49 (IMR3_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR3 Write Access Policy 48 (IMR3_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR3 Write Access Policy 47 (IMR3_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR3 Write Access Policy 46 (IMR3_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR3 Write Access Policy 45 (IMR3_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR3 Write Access Policy 44 (IMR3_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR3 Write Access Policy 43 (IMR3_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR3 Write Access Policy 42 (IMR3_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
41	0h RW	**IMR3 Write Access Policy 41 (IMR3_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR3 Write Access Policy 40 (IMR3_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR3 Write Access Policy 39 (IMR3_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR3 Write Access Policy 38 (IMR3_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR3 Write Access Policy 37 (IMR3_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR3 Write Access Policy 36 (IMR3_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR3 Write Access Policy 35 (IMR3_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR3 Write Access Policy 34 (IMR3_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR3 Write Access Policy 33 (IMR3_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR3 Write Access Policy 32 (IMR3_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR3 Write Access Policy 31 (IMR3_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR3 Write Access Policy 30 (IMR3_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR3 Write Access Policy 29 (IMR3_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR3 Write Access Policy 28 (IMR3_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR3 Write Access Policy 27 (IMR3_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR3 Write Access Policy 26 (IMR3_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR3 Write Access Policy 25 (IMR3_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR3 Write Access Policy 24 (IMR3_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR3 Write Access Policy 23 (IMR3_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
22	0h RO	**IMR3 Write Access Policy 22 (IMR3_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR3 Write Access Policy 21 (IMR3_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR3 Write Access Policy 20 (IMR3_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR3 Write Access Policy 19 (IMR3_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR3 Write Access Policy 18 (IMR3_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR3 Write Access Policy 17 (IMR3_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR3 Write Access Policy 16 (IMR3_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR3 Write Access Policy 15 (IMR3_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR3 Write Access Policy 14 (IMR3_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR3 Write Access Policy 13 (IMR3_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR3 Write Access Policy 12 (IMR3_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR3 Write Access Policy 11 (IMR3_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR3 Write Access Policy 10 (IMR3_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR3 Write Access Policy 9 (IMR3_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR3 Write Access Policy 8 (IMR3_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR3 Write Access Policy 7 (IMR3_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR3 Write Access Policy 6 (IMR3_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR3 Write Access Policy 5 (IMR3_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR3 Write Access Policy 4 (IMR3_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**IMR3 Write Access Policy 3 (IMR3_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR3 Write Access Policy 2 (IMR3_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR3 Write Access Policy 1 (IMR3_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR3 Write Access Policy 0 (IMR3_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR3 region, based on each agent's 6bit encoded SAI value.

3.308 IMR4 Base (B_CR_BIMR4BASE_0_0_0_MCHBAR) — Offset 68F0h

This register, along with IMR4MASK, IMR4RAC, and IMR4WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR4RAC and IMR4WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 68F0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR4 Base (IMR4_BASE):** Specifies bits 38:10 of the start address of IMR4 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR4MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR4 defined region.

3.309 IMR4 Mask (B_CR_BIMR4MASK_0_0_0_MCHBAR) — Offset 68F4h

This register, along with IMR4BASE, IMR4RAC, and IMR4WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or

from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR4RAC and IMR4WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 68F4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved
28:0	0h RW	**Mask 0 IMR4 Mask (IMR4_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR4BASE[28:0] value. A match indicates that the incoming address falls within the IMR4 region.

3.310 IMR4 Control Policy (B_CR_BIMR4CP_0_0_0_MCHBAR) — Offset 68F8h

This register controls the access policy to the Read Access Policy BIMR4RAC, Write Access Policy BIMR4WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 68F8h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
60	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
59	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
58	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
55	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
52	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
50	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
47	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
44	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR4 Control Policy (IMR4_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR4 Control Policy (IMR4_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
41	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
40	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
39	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
36	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
33	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
31	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR4 Control Policy (IMR4_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR4 Control Policy (IMR4_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
28	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
25	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR4 Control Policy (IMR4_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
22	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
21	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
20	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
17	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR4 Control Policy (IMR4_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
14	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
12	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
9	1h RW	**IMR4 Control Policy (IMR4_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
6	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
2	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
1	1h RW	**IMR4 Control Policy (IMR4_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR4 Control Policy (IMR4_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR4RAC, BIMR4WAC and BIMR4CP registers based on the value from each agent's 6bit SAI field.

3.311 IMR4 Read Access Policy (B_CR_BIMR4RAC_0_0_0_MCHBAR) — Offset 6900h

This register, along with IMR4BASE, IMR4MASK and IMR4WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR4. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6900h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR4 Read Access Policy 63 (IMR4_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR4 Read Access Policy 62 (IMR4_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR4 Read Access Policy 61 (IMR4_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR4 Read Access Policy 60 (IMR4_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR4 Read Access Policy 59 (IMR4_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR4 Read Access Policy 58 (IMR4_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR4 Read Access Policy 57 (IMR4_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR4 Read Access Policy 56 (IMR4_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
55	0h RW	**IMR4 Read Access Policy 55 (IMR4_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR4 Read Access Policy 54 (IMR4_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR4 Read Access Policy 53 (IMR4_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR4 Read Access Policy 52 (IMR4_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR4 Read Access Policy 51 (IMR4_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR4 Read Access Policy 50 (IMR4_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR4 Read Access Policy 49 (IMR4_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR4 Read Access Policy 48 (IMR4_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR4 Read Access Policy 47 (IMR4_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR4 Read Access Policy 46 (IMR4_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR4 Read Access Policy 45 (IMR4_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR4 Read Access Policy 44 (IMR4_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR4 Read Access Policy 43 (IMR4_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR4 Read Access Policy 42 (IMR4_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR4 Read Access Policy 41 (IMR4_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR4 Read Access Policy 40 (IMR4_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR4 Read Access Policy 39 (IMR4_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR4 Read Access Policy 38 (IMR4_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR4 Read Access Policy 37 (IMR4_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
36	0h RW	**IMR4 Read Access Policy 36 (IMR4_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR4 Read Access Policy 35 (IMR4_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR4 Read Access Policy 34 (IMR4_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR4 Read Access Policy 33 (IMR4_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR4 Read Access Policy 32 (IMR4_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR4 Read Access Policy 31 (IMR4_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR4 Read Access Policy 30 (IMR4_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR4 Read Access Policy 29 (IMR4_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR4 Read Access Policy 28 (IMR4_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR4 Read Access Policy 27 (IMR4_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR4 Read Access Policy 26 (IMR4_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR4 Read Access Policy 25 (IMR4_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR4 Read Access Policy 24 (IMR4_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR4 Read Access Policy 23 (IMR4_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR4 Read Access Policy 22 (IMR4_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR4 Read Access Policy 21 (IMR4_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR4 Read Access Policy 20 (IMR4_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR4 Read Access Policy 19 (IMR4_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR4 Read Access Policy 18 (IMR4_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**IMR4 Read Access Policy 17 (IMR4_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR4 Read Access Policy 16 (IMR4_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR4 Read Access Policy 15 (IMR4_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR4 Read Access Policy 14 (IMR4_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR4 Read Access Policy 13 (IMR4_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR4 Read Access Policy 12 (IMR4_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR4 Read Access Policy 11 (IMR4_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR4 Read Access Policy 10 (IMR4_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR4 Read Access Policy 9 (IMR4_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR4 Read Access Policy 8 (IMR4_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR4 Read Access Policy 7 (IMR4_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR4 Read Access Policy 6 (IMR4_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR4 Read Access Policy 5 (IMR4_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR4 Read Access Policy 4 (IMR4_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR4 Read Access Policy 3 (IMR4_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR4 Read Access Policy 2 (IMR4_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR4 Read Access Policy 1 (IMR4_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR4 Read Access Policy 0 (IMR4_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR4 region, based on each agent's 6bit encoded SAI value.

3.312 IMR4 Write Access Policy (B_CR_BIMR4WAC_0_0_0_MCHBAR) — Offset 6908h

This register, along with IMR4BASE, IMR4MASK and IMR4RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR4. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6908h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR4 Write Access Policy 63 (IMR4_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR4 Write Access Policy 62 (IMR4_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR4 Write Access Policy 61 (IMR4_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR4 Write Access Policy 60 (IMR4_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR4 Write Access Policy 59 (IMR4_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR4 Write Access Policy 58 (IMR4_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR4 Write Access Policy 57 (IMR4_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR4 Write Access Policy 56 (IMR4_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR4 Write Access Policy 55 (IMR4_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR4 Write Access Policy 54 (IMR4_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR4 Write Access Policy 53 (IMR4_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR4 Write Access Policy 52 (IMR4_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR4 Write Access Policy 51 (IMR4_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
50	0h RW	**IMR4 Write Access Policy 50 (IMR4_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR4 Write Access Policy 49 (IMR4_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR4 Write Access Policy 48 (IMR4_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR4 Write Access Policy 47 (IMR4_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR4 Write Access Policy 46 (IMR4_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR4 Write Access Policy 45 (IMR4_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR4 Write Access Policy 44 (IMR4_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR4 Write Access Policy 43 (IMR4_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR4 Write Access Policy 42 (IMR4_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR4 Write Access Policy 41 (IMR4_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR4 Write Access Policy 40 (IMR4_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR4 Write Access Policy 39 (IMR4_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR4 Write Access Policy 38 (IMR4_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR4 Write Access Policy 37 (IMR4_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR4 Write Access Policy 36 (IMR4_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR4 Write Access Policy 35 (IMR4_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR4 Write Access Policy 34 (IMR4_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR4 Write Access Policy 33 (IMR4_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR4 Write Access Policy 32 (IMR4_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**IMR4 Write Access Policy 31 (IMR4_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR4 Write Access Policy 30 (IMR4_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR4 Write Access Policy 29 (IMR4_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR4 Write Access Policy 28 (IMR4_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR4 Write Access Policy 27 (IMR4_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR4 Write Access Policy 26 (IMR4_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR4 Write Access Policy 25 (IMR4_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR4 Write Access Policy 24 (IMR4_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR4 Write Access Policy 23 (IMR4_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR4 Write Access Policy 22 (IMR4_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR4 Write Access Policy 21 (IMR4_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR4 Write Access Policy 20 (IMR4_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR4 Write Access Policy 19 (IMR4_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR4 Write Access Policy 18 (IMR4_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR4 Write Access Policy 17 (IMR4_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR4 Write Access Policy 16 (IMR4_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR4 Write Access Policy 15 (IMR4_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR4 Write Access Policy 14 (IMR4_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR4 Write Access Policy 13 (IMR4_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**IMR4 Write Access Policy 12 (IMR4_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR4 Write Access Policy 11 (IMR4_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR4 Write Access Policy 10 (IMR4_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR4 Write Access Policy 9 (IMR4_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR4 Write Access Policy 8 (IMR4_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR4 Write Access Policy 7 (IMR4_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR4 Write Access Policy 6 (IMR4_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR4 Write Access Policy 5 (IMR4_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR4 Write Access Policy 4 (IMR4_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR4 Write Access Policy 3 (IMR4_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR4 Write Access Policy 2 (IMR4_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR4 Write Access Policy 1 (IMR4_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR4 Write Access Policy 0 (IMR4_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR4 region, based on each agent's 6bit encoded SAI value.

3.313 IMR5 Base (B_CR_BIMR5BASE_0_0_0_MCHBAR) — Offset 6910h

This register, along with IMR5MASK, IMR5RAC, and IMR5WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR5RAC and IMR5WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6910h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR5 Base (IMR5_BASE):** Specifies bits 38:10 of the start address of IMR5 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR5MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR5 defined region.

3.314 IMR5 Mask (B_CR_BIMR5MASK_0_0_0_MCHBAR) — Offset 6914h

This register, along with IMR5BASE, IMR5RAC, and IMR5WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR5RAC and IMR5WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6914h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved
28:0	0h RW	**Mask 0 IMR5 Mask (IMR5_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR5BASE[28:0] value. A match indicates that the incoming address falls within the IMR5 region.

3.315 IMR5 Control Policy (B_CR_BIMR5CP_0_0_0_MCHBAR) — Offset 6918h

This register controls the access policy to the Read Access Policy BIMR5RAC, Write Access Policy BIMR5WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6918h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
60	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
59	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
58	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
55	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
52	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
50	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
47	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
44	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR5 Control Policy (IMR5_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR5 Control Policy (IMR5_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
41	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
40	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
39	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
36	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
33	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR5 Control Policy (IMR5_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR5 Control Policy (IMR5_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
28	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
25	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR5 Control Policy (IMR5_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
22	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
21	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
20	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
17	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR5 Control Policy (IMR5_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
14	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
9	1h RW	**IMR5 Control Policy (IMR5_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
6	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
3	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
2	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
1	1h RW	**IMR5 Control Policy (IMR5_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR5 Control Policy (IMR5_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR5RAC, BIMR5WAC and BIMR5CP registers based on the value from each agent's 6bit SAI field.

3.316 IMR5 Read Access Policy (B_CR_BIMR5RAC_0_0_0_MCHBAR) — Offset 6920h

This register, along with IMR5BASE, IMR5MASK and IMR5WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR5. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6920h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR5 Read Access Policy 63 (IMR5_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR5 Read Access Policy 62 (IMR5_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR5 Read Access Policy 61 (IMR5_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR5 Read Access Policy 60 (IMR5_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR5 Read Access Policy 59 (IMR5_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR5 Read Access Policy 58 (IMR5_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR5 Read Access Policy 57 (IMR5_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR5 Read Access Policy 56 (IMR5_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR5 Read Access Policy 55 (IMR5_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR5 Read Access Policy 54 (IMR5_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR5 Read Access Policy 53 (IMR5_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR5 Read Access Policy 52 (IMR5_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR5 Read Access Policy 51 (IMR5_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR5 Read Access Policy 50 (IMR5_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR5 Read Access Policy 49 (IMR5_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR5 Read Access Policy 48 (IMR5_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR5 Read Access Policy 47 (IMR5_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR5 Read Access Policy 46 (IMR5_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR5 Read Access Policy 45 (IMR5_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
44	0h RW	**IMR5 Read Access Policy 44 (IMR5_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR5 Read Access Policy 43 (IMR5_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR5 Read Access Policy 42 (IMR5_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR5 Read Access Policy 41 (IMR5_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR5 Read Access Policy 40 (IMR5_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR5 Read Access Policy 39 (IMR5_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR5 Read Access Policy 38 (IMR5_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR5 Read Access Policy 37 (IMR5_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR5 Read Access Policy 36 (IMR5_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR5 Read Access Policy 35 (IMR5_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR5 Read Access Policy 34 (IMR5_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR5 Read Access Policy 33 (IMR5_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR5 Read Access Policy 32 (IMR5_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR5 Read Access Policy 31 (IMR5_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR5 Read Access Policy 30 (IMR5_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR5 Read Access Policy 29 (IMR5_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR5 Read Access Policy 28 (IMR5_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR5 Read Access Policy 27 (IMR5_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR5 Read Access Policy 26 (IMR5_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**IMR5 Read Access Policy 25 (IMR5_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR5 Read Access Policy 24 (IMR5_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR5 Read Access Policy 23 (IMR5_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR5 Read Access Policy 22 (IMR5_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR5 Read Access Policy 21 (IMR5_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR5 Read Access Policy 20 (IMR5_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR5 Read Access Policy 19 (IMR5_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR5 Read Access Policy 18 (IMR5_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR5 Read Access Policy 17 (IMR5_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR5 Read Access Policy 16 (IMR5_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR5 Read Access Policy 15 (IMR5_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR5 Read Access Policy 14 (IMR5_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR5 Read Access Policy 13 (IMR5_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR5 Read Access Policy 12 (IMR5_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR5 Read Access Policy 11 (IMR5_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR5 Read Access Policy 10 (IMR5_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR5 Read Access Policy 9 (IMR5_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR5 Read Access Policy 8 (IMR5_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR5 Read Access Policy 7 (IMR5_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**IMR5 Read Access Policy 6 (IMR5_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR5 Read Access Policy 5 (IMR5_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR5 Read Access Policy 4 (IMR5_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR5 Read Access Policy 3 (IMR5_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR5 Read Access Policy 2 (IMR5_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR5 Read Access Policy 1 (IMR5_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR5 Read Access Policy 0 (IMR5_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR5 region, based on each agent's 6bit encoded SAI value.

3.317 IMR5 Write Access Policy (B_CR_BIMR5WAC_0_0_0_MCHBAR) — Offset 6928h

This register, along with IMR5BASE, IMR5MASK and IMR5RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR5. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6928h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR5 Write Access Policy 63 (IMR5_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR5 Write Access Policy 62 (IMR5_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR5 Write Access Policy 61 (IMR5_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR5 Write Access Policy 60 (IMR5_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR5 Write Access Policy 59 (IMR5_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
58	0h RO	**IMR5 Write Access Policy 58 (IMR5_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR5 Write Access Policy 57 (IMR5_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR5 Write Access Policy 56 (IMR5_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR5 Write Access Policy 55 (IMR5_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR5 Write Access Policy 54 (IMR5_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR5 Write Access Policy 53 (IMR5_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR5 Write Access Policy 52 (IMR5_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR5 Write Access Policy 51 (IMR5_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR5 Write Access Policy 50 (IMR5_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR5 Write Access Policy 49 (IMR5_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR5 Write Access Policy 48 (IMR5_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR5 Write Access Policy 47 (IMR5_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR5 Write Access Policy 46 (IMR5_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR5 Write Access Policy 45 (IMR5_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR5 Write Access Policy 44 (IMR5_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR5 Write Access Policy 43 (IMR5_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR5 Write Access Policy 42 (IMR5_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR5 Write Access Policy 41 (IMR5_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR5 Write Access Policy 40 (IMR5_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
39	0h RO	**IMR5 Write Access Policy 39 (IMR5_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR5 Write Access Policy 38 (IMR5_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR5 Write Access Policy 37 (IMR5_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR5 Write Access Policy 36 (IMR5_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR5 Write Access Policy 35 (IMR5_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR5 Write Access Policy 34 (IMR5_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR5 Write Access Policy 33 (IMR5_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR5 Write Access Policy 32 (IMR5_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR5 Write Access Policy 31 (IMR5_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR5 Write Access Policy 30 (IMR5_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR5 Write Access Policy 29 (IMR5_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR5 Write Access Policy 28 (IMR5_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR5 Write Access Policy 27 (IMR5_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR5 Write Access Policy 26 (IMR5_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR5 Write Access Policy 25 (IMR5_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR5 Write Access Policy 24 (IMR5_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR5 Write Access Policy 23 (IMR5_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR5 Write Access Policy 22 (IMR5_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR5 Write Access Policy 21 (IMR5_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**IMR5 Write Access Policy 20 (IMR5_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR5 Write Access Policy 19 (IMR5_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR5 Write Access Policy 18 (IMR5_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR5 Write Access Policy 17 (IMR5_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR5 Write Access Policy 16 (IMR5_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR5 Write Access Policy 15 (IMR5_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR5 Write Access Policy 14 (IMR5_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR5 Write Access Policy 13 (IMR5_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR5 Write Access Policy 12 (IMR5_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR5 Write Access Policy 11 (IMR5_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR5 Write Access Policy 10 (IMR5_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR5 Write Access Policy 9 (IMR5_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR5 Write Access Policy 8 (IMR5_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR5 Write Access Policy 7 (IMR5_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR5 Write Access Policy 6 (IMR5_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR5 Write Access Policy 5 (IMR5_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR5 Write Access Policy 4 (IMR5_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR5 Write Access Policy 3 (IMR5_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR5 Write Access Policy 2 (IMR5_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**IMR5 Write Access Policy 1 (IMR5_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR5 Write Access Policy 0 (IMR5_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR5 region, based on each agent's 6bit encoded SAI value.

3.318 IMR6 Base (B_CR_BIMR6BASE_0_0_0_MCHBAR) — Offset 6930h

This register, along with IMR6MASK, IMR6RAC, and IMR6WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR6RAC and IMR6WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6930h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR6 Base (IMR6_BASE):** Specifies bits 38:10 of the start address of IMR6 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR6MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR6 defined region.

3.319 IMR6 Mask (B_CR_BIMR6MASK_0_0_0_MCHBAR) — Offset 6934h

This register, along with IMR6BASE, IMR6RAC, and IMR6WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR6RAC and IMR6WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6934h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved
28:0	0h RW	**Mask 0 IMR6 Mask (IMR6_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR6BASE[28:0] value. A match indicates that the incoming address falls within the IMR6 region.

3.320 IMR6 Control Policy (B_CR_BIMR6CP_0_0_0_MCHBAR) — Offset 6938h

This register controls the access policy to the Read Access Policy BIMR6RAC, Write Access Policy BIMR6WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6938h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
60	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
59	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
58	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
55	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
52	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
50	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
47	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
44	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR6 Control Policy (IMR6_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR6 Control Policy (IMR6_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
41	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
40	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
39	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
36	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
33	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
31	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR6 Control Policy (IMR6_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR6 Control Policy (IMR6_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
28	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
25	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR6 Control Policy (IMR6_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
22	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
21	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
17	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR6 Control Policy (IMR6_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
14	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
12	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
9	1h RW	**IMR6 Control Policy (IMR6_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
6	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
3	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
2	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
1	1h RW	**IMR6 Control Policy (IMR6_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR6 Control Policy (IMR6_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR6RAC, BIMR6WAC and BIMR6CP registers based on the value from each agent's 6bit SAI field.

3.321 IMR6 Read Access Policy (B_CR_BIMR6RAC_0_0_0_MCHBAR) — Offset 6940h

This register, along with IMR6BASE, IMR6MASK and IMR6WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR6. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6940h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR6 Read Access Policy 63 (IMR6_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR6 Read Access Policy 62 (IMR6_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR6 Read Access Policy 61 (IMR6_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR6 Read Access Policy 60 (IMR6_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR6 Read Access Policy 59 (IMR6_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR6 Read Access Policy 58 (IMR6_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR6 Read Access Policy 57 (IMR6_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR6 Read Access Policy 56 (IMR6_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR6 Read Access Policy 55 (IMR6_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR6 Read Access Policy 54 (IMR6_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
53	0h RO	**IMR6 Read Access Policy 53 (IMR6_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR6 Read Access Policy 52 (IMR6_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR6 Read Access Policy 51 (IMR6_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR6 Read Access Policy 50 (IMR6_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR6 Read Access Policy 49 (IMR6_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR6 Read Access Policy 48 (IMR6_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR6 Read Access Policy 47 (IMR6_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR6 Read Access Policy 46 (IMR6_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR6 Read Access Policy 45 (IMR6_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR6 Read Access Policy 44 (IMR6_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR6 Read Access Policy 43 (IMR6_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR6 Read Access Policy 42 (IMR6_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR6 Read Access Policy 41 (IMR6_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR6 Read Access Policy 40 (IMR6_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR6 Read Access Policy 39 (IMR6_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR6 Read Access Policy 38 (IMR6_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR6 Read Access Policy 37 (IMR6_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR6 Read Access Policy 36 (IMR6_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR6 Read Access Policy 35 (IMR6_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
34	0h RW	**IMR6 Read Access Policy 34 (IMR6_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR6 Read Access Policy 33 (IMR6_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR6 Read Access Policy 32 (IMR6_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR6 Read Access Policy 31 (IMR6_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR6 Read Access Policy 30 (IMR6_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR6 Read Access Policy 29 (IMR6_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR6 Read Access Policy 28 (IMR6_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR6 Read Access Policy 27 (IMR6_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR6 Read Access Policy 26 (IMR6_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR6 Read Access Policy 25 (IMR6_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR6 Read Access Policy 24 (IMR6_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR6 Read Access Policy 23 (IMR6_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR6 Read Access Policy 22 (IMR6_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR6 Read Access Policy 21 (IMR6_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR6 Read Access Policy 20 (IMR6_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR6 Read Access Policy 19 (IMR6_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR6 Read Access Policy 18 (IMR6_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR6 Read Access Policy 17 (IMR6_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR6 Read Access Policy 16 (IMR6_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**IMR6 Read Access Policy 15 (IMR6_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR6 Read Access Policy 14 (IMR6_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR6 Read Access Policy 13 (IMR6_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR6 Read Access Policy 12 (IMR6_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR6 Read Access Policy 11 (IMR6_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR6 Read Access Policy 10 (IMR6_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR6 Read Access Policy 9 (IMR6_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR6 Read Access Policy 8 (IMR6_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR6 Read Access Policy 7 (IMR6_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR6 Read Access Policy 6 (IMR6_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR6 Read Access Policy 5 (IMR6_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR6 Read Access Policy 4 (IMR6_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR6 Read Access Policy 3 (IMR6_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR6 Read Access Policy 2 (IMR6_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR6 Read Access Policy 1 (IMR6_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR6 Read Access Policy 0 (IMR6_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR6 region, based on each agent's 6bit encoded SAI value.

3.322 IMR6 Write Access Policy (B_CR_BIMR6WAC_0_0_0_MCHBAR) — Offset 6948h

This register, along with IMR6BASE, IMR6MASK and IMR6RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR6. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6948h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR6 Write Access Policy 63 (IMR6_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR6 Write Access Policy 62 (IMR6_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR6 Write Access Policy 61 (IMR6_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR6 Write Access Policy 60 (IMR6_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR6 Write Access Policy 59 (IMR6_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR6 Write Access Policy 58 (IMR6_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR6 Write Access Policy 57 (IMR6_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR6 Write Access Policy 56 (IMR6_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR6 Write Access Policy 55 (IMR6_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR6 Write Access Policy 54 (IMR6_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR6 Write Access Policy 53 (IMR6_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR6 Write Access Policy 52 (IMR6_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR6 Write Access Policy 51 (IMR6_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
50	0h RW	**IMR6 Write Access Policy 50 (IMR6_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR6 Write Access Policy 49 (IMR6_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR6 Write Access Policy 48 (IMR6_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR6 Write Access Policy 47 (IMR6_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR6 Write Access Policy 46 (IMR6_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR6 Write Access Policy 45 (IMR6_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR6 Write Access Policy 44 (IMR6_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR6 Write Access Policy 43 (IMR6_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR6 Write Access Policy 42 (IMR6_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR6 Write Access Policy 41 (IMR6_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR6 Write Access Policy 40 (IMR6_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR6 Write Access Policy 39 (IMR6_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR6 Write Access Policy 38 (IMR6_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR6 Write Access Policy 37 (IMR6_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR6 Write Access Policy 36 (IMR6_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR6 Write Access Policy 35 (IMR6_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR6 Write Access Policy 34 (IMR6_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR6 Write Access Policy 33 (IMR6_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR6 Write Access Policy 32 (IMR6_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**IMR6 Write Access Policy 31 (IMR6_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR6 Write Access Policy 30 (IMR6_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR6 Write Access Policy 29 (IMR6_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR6 Write Access Policy 28 (IMR6_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR6 Write Access Policy 27 (IMR6_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR6 Write Access Policy 26 (IMR6_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR6 Write Access Policy 25 (IMR6_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR6 Write Access Policy 24 (IMR6_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR6 Write Access Policy 23 (IMR6_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR6 Write Access Policy 22 (IMR6_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR6 Write Access Policy 21 (IMR6_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR6 Write Access Policy 20 (IMR6_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR6 Write Access Policy 19 (IMR6_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR6 Write Access Policy 18 (IMR6_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR6 Write Access Policy 17 (IMR6_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR6 Write Access Policy 16 (IMR6_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR6 Write Access Policy 15 (IMR6_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR6 Write Access Policy 14 (IMR6_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR6 Write Access Policy 13 (IMR6_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**IMR6 Write Access Policy 12 (IMR6_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR6 Write Access Policy 11 (IMR6_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR6 Write Access Policy 10 (IMR6_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR6 Write Access Policy 9 (IMR6_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR6 Write Access Policy 8 (IMR6_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR6 Write Access Policy 7 (IMR6_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR6 Write Access Policy 6 (IMR6_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR6 Write Access Policy 5 (IMR6_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR6 Write Access Policy 4 (IMR6_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR6 Write Access Policy 3 (IMR6_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR6 Write Access Policy 2 (IMR6_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR6 Write Access Policy 1 (IMR6_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR6 Write Access Policy 0 (IMR6_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR6 region, based on each agent's 6bit encoded SAI value.

3.323 IMR7 Base (B_CR_BIMR7BASE_0_0_0_MCHBAR) — Offset 6950h

This register, along with IMR7MASK, IMR7RAC, and IMR7WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR7RAC and IMR7WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6950h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR7 Base (IMR7_BASE):** Specifies bits 38:10 of the start address of IMR7 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR7MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR7 defined region.

3.324 IMR7 Mask (B_CR_BIMR7MASK_0_0_0_MCHBAR) — Offset 6954h

This register, along with IMR7BASE, IMR7RAC, and IMR7WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR7RAC and IMR7WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6954h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved
28:0	0h RW	**Mask 0 IMR7 Mask (IMR7_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR7BASE[28:0] value. A match indicates that the incoming address falls within the IMR7 region.

3.325 IMR7 Control Policy (B_CR_BIMR7CP_0_0_0_MCHBAR) — Offset 6958h

This register controls the access policy to the Read Access Policy BIMR7RAC, Write Access Policy BIMR7WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6958h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
60	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
59	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
58	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
55	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
52	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
50	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
47	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
44	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR7 Control Policy (IMR7_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR7 Control Policy (IMR7_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
41	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
40	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
39	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
36	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
33	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR7 Control Policy (IMR7_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR7 Control Policy (IMR7_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
28	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
25	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR7 Control Policy (IMR7_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
22	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
21	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
20	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
17	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR7 Control Policy (IMR7_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
14	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
9	1h RW	**IMR7 Control Policy (IMR7_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
6	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
3	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
2	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
1	1h RW	**IMR7 Control Policy (IMR7_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR7 Control Policy (IMR7_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR7RAC, BIMR7WAC and BIMR7CP registers based on the value from each agent's 6bit SAI field.

3.326 IMR7 Read Access Policy (B_CR_BIMR7RAC_0_0_0_MCHBAR) — Offset 6960h

This register, along with IMR7BASE, IMR7MASK and IMR7WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR7. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6960h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR7 Read Access Policy 63 (IMR7_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR7 Read Access Policy 62 (IMR7_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR7 Read Access Policy 61 (IMR7_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR7 Read Access Policy 60 (IMR7_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR7 Read Access Policy 59 (IMR7_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR7 Read Access Policy 58 (IMR7_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR7 Read Access Policy 57 (IMR7_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR7 Read Access Policy 56 (IMR7_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR7 Read Access Policy 55 (IMR7_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR7 Read Access Policy 54 (IMR7_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR7 Read Access Policy 53 (IMR7_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR7 Read Access Policy 52 (IMR7_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR7 Read Access Policy 51 (IMR7_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR7 Read Access Policy 50 (IMR7_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR7 Read Access Policy 49 (IMR7_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR7 Read Access Policy 48 (IMR7_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR7 Read Access Policy 47 (IMR7_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR7 Read Access Policy 46 (IMR7_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR7 Read Access Policy 45 (IMR7_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
44	0h RW	**IMR7 Read Access Policy 44 (IMR7_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR7 Read Access Policy 43 (IMR7_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR7 Read Access Policy 42 (IMR7_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR7 Read Access Policy 41 (IMR7_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR7 Read Access Policy 40 (IMR7_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR7 Read Access Policy 39 (IMR7_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR7 Read Access Policy 38 (IMR7_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR7 Read Access Policy 37 (IMR7_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR7 Read Access Policy 36 (IMR7_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR7 Read Access Policy 35 (IMR7_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR7 Read Access Policy 34 (IMR7_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR7 Read Access Policy 33 (IMR7_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR7 Read Access Policy 32 (IMR7_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR7 Read Access Policy 31 (IMR7_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR7 Read Access Policy 30 (IMR7_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR7 Read Access Policy 29 (IMR7_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR7 Read Access Policy 28 (IMR7_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR7 Read Access Policy 27 (IMR7_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR7 Read Access Policy 26 (IMR7_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**IMR7 Read Access Policy 25 (IMR7_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR7 Read Access Policy 24 (IMR7_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR7 Read Access Policy 23 (IMR7_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR7 Read Access Policy 22 (IMR7_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR7 Read Access Policy 21 (IMR7_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR7 Read Access Policy 20 (IMR7_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR7 Read Access Policy 19 (IMR7_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR7 Read Access Policy 18 (IMR7_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR7 Read Access Policy 17 (IMR7_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR7 Read Access Policy 16 (IMR7_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR7 Read Access Policy 15 (IMR7_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR7 Read Access Policy 14 (IMR7_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR7 Read Access Policy 13 (IMR7_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR7 Read Access Policy 12 (IMR7_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR7 Read Access Policy 11 (IMR7_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR7 Read Access Policy 10 (IMR7_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR7 Read Access Policy 9 (IMR7_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR7 Read Access Policy 8 (IMR7_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR7 Read Access Policy 7 (IMR7_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**IMR7 Read Access Policy 6 (IMR7_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR7 Read Access Policy 5 (IMR7_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR7 Read Access Policy 4 (IMR7_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR7 Read Access Policy 3 (IMR7_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR7 Read Access Policy 2 (IMR7_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR7 Read Access Policy 1 (IMR7_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR7 Read Access Policy 0 (IMR7_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR7 region, based on each agent's 6bit encoded SAI value.

3.327 IMR7 Write Access Policy (B_CR_BIMR7WAC_0_0_0_MCHBAR) — Offset 6968h

This register, along with IMR7BASE, IMR7MASK and IMR7RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR7. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6968h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR7 Write Access Policy 63 (IMR7_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR7 Write Access Policy 62 (IMR7_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR7 Write Access Policy 61 (IMR7_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR7 Write Access Policy 60 (IMR7_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR7 Write Access Policy 59 (IMR7_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
58	0h RO	**IMR7 Write Access Policy 58 (IMR7_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR7 Write Access Policy 57 (IMR7_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR7 Write Access Policy 56 (IMR7_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR7 Write Access Policy 55 (IMR7_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR7 Write Access Policy 54 (IMR7_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR7 Write Access Policy 53 (IMR7_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR7 Write Access Policy 52 (IMR7_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR7 Write Access Policy 51 (IMR7_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR7 Write Access Policy 50 (IMR7_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR7 Write Access Policy 49 (IMR7_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR7 Write Access Policy 48 (IMR7_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR7 Write Access Policy 47 (IMR7_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR7 Write Access Policy 46 (IMR7_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR7 Write Access Policy 45 (IMR7_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR7 Write Access Policy 44 (IMR7_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR7 Write Access Policy 43 (IMR7_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR7 Write Access Policy 42 (IMR7_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR7 Write Access Policy 41 (IMR7_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR7 Write Access Policy 40 (IMR7_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
39	0h RO	**IMR7 Write Access Policy 39 (IMR7_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR7 Write Access Policy 38 (IMR7_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR7 Write Access Policy 37 (IMR7_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR7 Write Access Policy 36 (IMR7_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR7 Write Access Policy 35 (IMR7_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR7 Write Access Policy 34 (IMR7_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR7 Write Access Policy 33 (IMR7_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR7 Write Access Policy 32 (IMR7_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR7 Write Access Policy 31 (IMR7_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR7 Write Access Policy 30 (IMR7_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR7 Write Access Policy 29 (IMR7_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR7 Write Access Policy 28 (IMR7_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR7 Write Access Policy 27 (IMR7_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR7 Write Access Policy 26 (IMR7_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR7 Write Access Policy 25 (IMR7_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR7 Write Access Policy 24 (IMR7_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR7 Write Access Policy 23 (IMR7_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR7 Write Access Policy 22 (IMR7_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR7 Write Access Policy 21 (IMR7_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**IMR7 Write Access Policy 20 (IMR7_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR7 Write Access Policy 19 (IMR7_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR7 Write Access Policy 18 (IMR7_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR7 Write Access Policy 17 (IMR7_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR7 Write Access Policy 16 (IMR7_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR7 Write Access Policy 15 (IMR7_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR7 Write Access Policy 14 (IMR7_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR7 Write Access Policy 13 (IMR7_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR7 Write Access Policy 12 (IMR7_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR7 Write Access Policy 11 (IMR7_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR7 Write Access Policy 10 (IMR7_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR7 Write Access Policy 9 (IMR7_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR7 Write Access Policy 8 (IMR7_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR7 Write Access Policy 7 (IMR7_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR7 Write Access Policy 6 (IMR7_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR7 Write Access Policy 5 (IMR7_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR7 Write Access Policy 4 (IMR7_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR7 Write Access Policy 3 (IMR7_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR7 Write Access Policy 2 (IMR7_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**IMR7 Write Access Policy 1 (IMR7_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR7 Write Access Policy 0 (IMR7_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR7 region, based on each agent's 6bit encoded SAI value.

3.328 IMR8 Base (B_CR_BIMR8BASE_0_0_0_MCHBAR) — Offset 6970h

This register, along with IMR8MASK, IMR8RAC, and IMR8WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR8RAC and IMR8WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6970h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR8 Base (IMR8_BASE):** Specifies bits 38:10 of the start address of IMR8 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR8MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR8 defined region.

3.329 IMR8 Mask (B_CR_BIMR8MASK_0_0_0_MCHBAR) — Offset 6974h

This register, along with IMR8BASE, IMR8RAC, and IMR8WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR8RAC and IMR8WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6974h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved
28:0	0h RW	**Mask 0 IMR8 Mask (IMR8_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR8BASE[28:0] value. A match indicates that the incoming address falls within the IMR8 region.

3.330 IMR8 Control Policy (B_CR_BIMR8CP_0_0_0_MCHBAR) — Offset 6978h

This register controls the access policy to the Read Access Policy BIMR8RAC, Write Access Policy BIMR8WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6978h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
60	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
59	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
58	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
55	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
52	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
50	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
47	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
44	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR8 Control Policy (IMR8_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR8 Control Policy (IMR8_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
41	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
40	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
39	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
36	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
33	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
31	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR8 Control Policy (IMR8_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR8 Control Policy (IMR8_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
28	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
25	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR8 Control Policy (IMR8_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
22	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
21	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
17	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR8 Control Policy (IMR8_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
14	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
12	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
9	1h RW	**IMR8 Control Policy (IMR8_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
6	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
3	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
2	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
1	1h RW	**IMR8 Control Policy (IMR8_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR8 Control Policy (IMR8_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR8RAC, BIMR8WAC and BIMR8CP registers based on the value from each agent's 6bit SAI field.

3.331 IMR8 Read Access Policy (B_CR_BIMR8RAC_0_0_0_MCHBAR) — Offset 6980h

This register, along with IMR8BASE, IMR8MASK and IMR8WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR8. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6980h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR8 Read Access Policy 63 (IMR8_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR8 Read Access Policy 62 (IMR8_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR8 Read Access Policy 61 (IMR8_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR8 Read Access Policy 60 (IMR8_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR8 Read Access Policy 59 (IMR8_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR8 Read Access Policy 58 (IMR8_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR8 Read Access Policy 57 (IMR8_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR8 Read Access Policy 56 (IMR8_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR8 Read Access Policy 55 (IMR8_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR8 Read Access Policy 54 (IMR8_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
53	0h RO	**IMR8 Read Access Policy 53 (IMR8_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR8 Read Access Policy 52 (IMR8_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR8 Read Access Policy 51 (IMR8_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR8 Read Access Policy 50 (IMR8_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR8 Read Access Policy 49 (IMR8_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR8 Read Access Policy 48 (IMR8_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR8 Read Access Policy 47 (IMR8_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR8 Read Access Policy 46 (IMR8_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR8 Read Access Policy 45 (IMR8_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR8 Read Access Policy 44 (IMR8_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR8 Read Access Policy 43 (IMR8_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR8 Read Access Policy 42 (IMR8_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR8 Read Access Policy 41 (IMR8_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR8 Read Access Policy 40 (IMR8_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR8 Read Access Policy 39 (IMR8_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR8 Read Access Policy 38 (IMR8_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR8 Read Access Policy 37 (IMR8_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR8 Read Access Policy 36 (IMR8_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR8 Read Access Policy 35 (IMR8_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
34	0h RW	**IMR8 Read Access Policy 34 (IMR8_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR8 Read Access Policy 33 (IMR8_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR8 Read Access Policy 32 (IMR8_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR8 Read Access Policy 31 (IMR8_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR8 Read Access Policy 30 (IMR8_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR8 Read Access Policy 29 (IMR8_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR8 Read Access Policy 28 (IMR8_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR8 Read Access Policy 27 (IMR8_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR8 Read Access Policy 26 (IMR8_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR8 Read Access Policy 25 (IMR8_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR8 Read Access Policy 24 (IMR8_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR8 Read Access Policy 23 (IMR8_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR8 Read Access Policy 22 (IMR8_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR8 Read Access Policy 21 (IMR8_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR8 Read Access Policy 20 (IMR8_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR8 Read Access Policy 19 (IMR8_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR8 Read Access Policy 18 (IMR8_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR8 Read Access Policy 17 (IMR8_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR8 Read Access Policy 16 (IMR8_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**IMR8 Read Access Policy 15 (IMR8_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR8 Read Access Policy 14 (IMR8_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR8 Read Access Policy 13 (IMR8_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR8 Read Access Policy 12 (IMR8_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR8 Read Access Policy 11 (IMR8_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR8 Read Access Policy 10 (IMR8_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR8 Read Access Policy 9 (IMR8_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR8 Read Access Policy 8 (IMR8_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR8 Read Access Policy 7 (IMR8_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR8 Read Access Policy 6 (IMR8_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR8 Read Access Policy 5 (IMR8_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR8 Read Access Policy 4 (IMR8_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR8 Read Access Policy 3 (IMR8_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR8 Read Access Policy 2 (IMR8_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR8 Read Access Policy 1 (IMR8_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR8 Read Access Policy 0 (IMR8_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR8 region, based on each agent's 6bit encoded SAI value.

3.332 IMR8 Write Access Policy (B_CR_BIMR8WAC_0_0_0_MCHBAR) — Offset 6988h

This register, along with IMR8BASE, IMR8MASK and IMR8RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR8. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6988h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR8 Write Access Policy 63 (IMR8_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR8 Write Access Policy 62 (IMR8_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR8 Write Access Policy 61 (IMR8_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR8 Write Access Policy 60 (IMR8_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR8 Write Access Policy 59 (IMR8_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR8 Write Access Policy 58 (IMR8_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR8 Write Access Policy 57 (IMR8_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR8 Write Access Policy 56 (IMR8_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR8 Write Access Policy 55 (IMR8_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR8 Write Access Policy 54 (IMR8_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR8 Write Access Policy 53 (IMR8_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR8 Write Access Policy 52 (IMR8_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR8 Write Access Policy 51 (IMR8_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
50	0h RW	**IMR8 Write Access Policy 50 (IMR8_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR8 Write Access Policy 49 (IMR8_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR8 Write Access Policy 48 (IMR8_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR8 Write Access Policy 47 (IMR8_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR8 Write Access Policy 46 (IMR8_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR8 Write Access Policy 45 (IMR8_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR8 Write Access Policy 44 (IMR8_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR8 Write Access Policy 43 (IMR8_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR8 Write Access Policy 42 (IMR8_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR8 Write Access Policy 41 (IMR8_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR8 Write Access Policy 40 (IMR8_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR8 Write Access Policy 39 (IMR8_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR8 Write Access Policy 38 (IMR8_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR8 Write Access Policy 37 (IMR8_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR8 Write Access Policy 36 (IMR8_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR8 Write Access Policy 35 (IMR8_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR8 Write Access Policy 34 (IMR8_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR8 Write Access Policy 33 (IMR8_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR8 Write Access Policy 32 (IMR8_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**IMR8 Write Access Policy 31 (IMR8_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR8 Write Access Policy 30 (IMR8_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR8 Write Access Policy 29 (IMR8_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR8 Write Access Policy 28 (IMR8_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR8 Write Access Policy 27 (IMR8_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR8 Write Access Policy 26 (IMR8_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR8 Write Access Policy 25 (IMR8_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR8 Write Access Policy 24 (IMR8_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR8 Write Access Policy 23 (IMR8_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR8 Write Access Policy 22 (IMR8_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR8 Write Access Policy 21 (IMR8_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR8 Write Access Policy 20 (IMR8_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR8 Write Access Policy 19 (IMR8_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR8 Write Access Policy 18 (IMR8_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR8 Write Access Policy 17 (IMR8_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR8 Write Access Policy 16 (IMR8_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR8 Write Access Policy 15 (IMR8_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR8 Write Access Policy 14 (IMR8_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR8 Write Access Policy 13 (IMR8_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**IMR8 Write Access Policy 12 (IMR8_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR8 Write Access Policy 11 (IMR8_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR8 Write Access Policy 10 (IMR8_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR8 Write Access Policy 9 (IMR8_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR8 Write Access Policy 8 (IMR8_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR8 Write Access Policy 7 (IMR8_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR8 Write Access Policy 6 (IMR8_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR8 Write Access Policy 5 (IMR8_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR8 Write Access Policy 4 (IMR8_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR8 Write Access Policy 3 (IMR8_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR8 Write Access Policy 2 (IMR8_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR8 Write Access Policy 1 (IMR8_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR8 Write Access Policy 0 (IMR8_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR8 region, based on each agent's 6bit encoded SAI value.

3.333 IMR9 Base (B_CR_BIMR9BASE_0_0_0_MCHBAR) — Offset 6990h

This register, along with IMR9MASK, IMR9RAC, and IMR9WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR9RAC and IMR9WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6990h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR9 Base (IMR9_BASE):** Specifies bits 38:10 of the start address of IMR9 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR9MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR9 defined region.

3.334 IMR9 Mask (B_CR_BIMR9MASK_0_0_0_MCHBAR) — Offset 6994h

This register, along with IMR9BASE, IMR9RAC, and IMR9WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR9RAC and IMR9WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6994h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved
28:0	0h RW	**Mask 0 IMR9 Mask (IMR9_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR9BASE[28:0] value. A match indicates that the incoming address falls within the IMR9 region.

3.335 IMR9 Control Policy (B_CR_BIMR9CP_0_0_0_MCHBAR) — Offset 6998h

This register controls the access policy to the Read Access Policy BIMR9RAC, Write Access Policy BIMR9WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6998h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
60	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
59	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
58	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
55	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
52	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
50	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
47	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
44	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR9 Control Policy (IMR9_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR9 Control Policy (IMR9_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
41	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
40	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
39	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
36	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
33	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR9 Control Policy (IMR9_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR9 Control Policy (IMR9_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
28	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
25	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR9 Control Policy (IMR9_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
22	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
21	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
20	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
17	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR9 Control Policy (IMR9_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
14	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
9	1h RW	**IMR9 Control Policy (IMR9_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
6	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
3	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
2	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
1	1h RW	**IMR9 Control Policy (IMR9_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR9 Control Policy (IMR9_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR9RAC, BIMR9WAC and BIMR9CP registers based on the value from each agent's 6bit SAI field.

3.336 IMR9 Read Access Policy (B_CR_BIMR9RAC_0_0_0_MCHBAR) — Offset 69A0h

This register, along with IMR9BASE, IMR9MASK and IMR9WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR9. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 69A0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR9 Read Access Policy 63 (IMR9_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR9 Read Access Policy 62 (IMR9_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR9 Read Access Policy 61 (IMR9_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR9 Read Access Policy 60 (IMR9_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR9 Read Access Policy 59 (IMR9_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR9 Read Access Policy 58 (IMR9_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR9 Read Access Policy 57 (IMR9_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR9 Read Access Policy 56 (IMR9_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR9 Read Access Policy 55 (IMR9_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR9 Read Access Policy 54 (IMR9_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR9 Read Access Policy 53 (IMR9_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR9 Read Access Policy 52 (IMR9_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR9 Read Access Policy 51 (IMR9_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR9 Read Access Policy 50 (IMR9_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR9 Read Access Policy 49 (IMR9_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR9 Read Access Policy 48 (IMR9_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR9 Read Access Policy 47 (IMR9_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR9 Read Access Policy 46 (IMR9_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR9 Read Access Policy 45 (IMR9_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
44	0h RW	**IMR9 Read Access Policy 44 (IMR9_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR9 Read Access Policy 43 (IMR9_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR9 Read Access Policy 42 (IMR9_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR9 Read Access Policy 41 (IMR9_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR9 Read Access Policy 40 (IMR9_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR9 Read Access Policy 39 (IMR9_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR9 Read Access Policy 38 (IMR9_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR9 Read Access Policy 37 (IMR9_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR9 Read Access Policy 36 (IMR9_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR9 Read Access Policy 35 (IMR9_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR9 Read Access Policy 34 (IMR9_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR9 Read Access Policy 33 (IMR9_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR9 Read Access Policy 32 (IMR9_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR9 Read Access Policy 31 (IMR9_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR9 Read Access Policy 30 (IMR9_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR9 Read Access Policy 29 (IMR9_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR9 Read Access Policy 28 (IMR9_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR9 Read Access Policy 27 (IMR9_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR9 Read Access Policy 26 (IMR9_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**IMR9 Read Access Policy 25 (IMR9_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR9 Read Access Policy 24 (IMR9_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR9 Read Access Policy 23 (IMR9_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR9 Read Access Policy 22 (IMR9_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR9 Read Access Policy 21 (IMR9_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR9 Read Access Policy 20 (IMR9_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR9 Read Access Policy 19 (IMR9_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR9 Read Access Policy 18 (IMR9_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR9 Read Access Policy 17 (IMR9_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR9 Read Access Policy 16 (IMR9_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR9 Read Access Policy 15 (IMR9_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR9 Read Access Policy 14 (IMR9_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR9 Read Access Policy 13 (IMR9_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR9 Read Access Policy 12 (IMR9_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR9 Read Access Policy 11 (IMR9_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR9 Read Access Policy 10 (IMR9_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR9 Read Access Policy 9 (IMR9_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR9 Read Access Policy 8 (IMR9_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR9 Read Access Policy 7 (IMR9_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**IMR9 Read Access Policy 6 (IMR9_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR9 Read Access Policy 5 (IMR9_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR9 Read Access Policy 4 (IMR9_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR9 Read Access Policy 3 (IMR9_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR9 Read Access Policy 2 (IMR9_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR9 Read Access Policy 1 (IMR9_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR9 Read Access Policy 0 (IMR9_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR9 region, based on each agent's 6bit encoded SAI value.

3.337 IMR9 Write Access Policy (B_CR_BIMR9WAC_0_0_0_MCHBAR) — Offset 69A8h

This register, along with IMR9BASE, IMR9MASK and IMR9RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR9. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 69A8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR9 Write Access Policy 63 (IMR9_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR9 Write Access Policy 62 (IMR9_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR9 Write Access Policy 61 (IMR9_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR9 Write Access Policy 60 (IMR9_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR9 Write Access Policy 59 (IMR9_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
58	0h RO	**IMR9 Write Access Policy 58 (IMR9_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR9 Write Access Policy 57 (IMR9_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR9 Write Access Policy 56 (IMR9_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR9 Write Access Policy 55 (IMR9_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR9 Write Access Policy 54 (IMR9_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR9 Write Access Policy 53 (IMR9_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR9 Write Access Policy 52 (IMR9_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR9 Write Access Policy 51 (IMR9_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR9 Write Access Policy 50 (IMR9_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR9 Write Access Policy 49 (IMR9_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR9 Write Access Policy 48 (IMR9_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR9 Write Access Policy 47 (IMR9_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR9 Write Access Policy 46 (IMR9_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR9 Write Access Policy 45 (IMR9_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR9 Write Access Policy 44 (IMR9_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR9 Write Access Policy 43 (IMR9_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR9 Write Access Policy 42 (IMR9_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR9 Write Access Policy 41 (IMR9_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR9 Write Access Policy 40 (IMR9_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
39	0h RO	**IMR9 Write Access Policy 39 (IMR9_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR9 Write Access Policy 38 (IMR9_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR9 Write Access Policy 37 (IMR9_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR9 Write Access Policy 36 (IMR9_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR9 Write Access Policy 35 (IMR9_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR9 Write Access Policy 34 (IMR9_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR9 Write Access Policy 33 (IMR9_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR9 Write Access Policy 32 (IMR9_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR9 Write Access Policy 31 (IMR9_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR9 Write Access Policy 30 (IMR9_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR9 Write Access Policy 29 (IMR9_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR9 Write Access Policy 28 (IMR9_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR9 Write Access Policy 27 (IMR9_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR9 Write Access Policy 26 (IMR9_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR9 Write Access Policy 25 (IMR9_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR9 Write Access Policy 24 (IMR9_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR9 Write Access Policy 23 (IMR9_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR9 Write Access Policy 22 (IMR9_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR9 Write Access Policy 21 (IMR9_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**IMR9 Write Access Policy 20 (IMR9_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR9 Write Access Policy 19 (IMR9_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR9 Write Access Policy 18 (IMR9_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR9 Write Access Policy 17 (IMR9_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR9 Write Access Policy 16 (IMR9_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR9 Write Access Policy 15 (IMR9_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR9 Write Access Policy 14 (IMR9_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR9 Write Access Policy 13 (IMR9_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR9 Write Access Policy 12 (IMR9_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR9 Write Access Policy 11 (IMR9_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR9 Write Access Policy 10 (IMR9_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR9 Write Access Policy 9 (IMR9_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR9 Write Access Policy 8 (IMR9_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR9 Write Access Policy 7 (IMR9_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR9 Write Access Policy 6 (IMR9_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR9 Write Access Policy 5 (IMR9_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR9 Write Access Policy 4 (IMR9_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR9 Write Access Policy 3 (IMR9_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR9 Write Access Policy 2 (IMR9_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**IMR9 Write Access Policy 1 (IMR9_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR9 Write Access Policy 0 (IMR9_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR9 region, based on each agent's 6bit encoded SAI value.

3.338 IMR10 Base (B_CR_BIMR10BASE_0_0_0_MCHBAR) — Offset 69B0h

This register, along with IMR10MASK, IMR10RAC, and IMR10WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR10RAC and IMR10WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 69B0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR10 Base (IMR10_BASE):** Specifies bits 38:10 of the start address of IMR10 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR10MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR10 defined region.

3.339 IMR10 Mask (B_CR_BIMR10MASK_0_0_0_MCHBAR) — Offset 69B4h

This register, along with IMR10BASE, IMR10RAC, and IMR10WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR10RAC and IMR10WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 69B4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved
28:0	0h RW	**Mask 0 IMR10 Mask (IMR10_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR10BASE[28:0] value. A match indicates that the incoming address falls within the IMR10 region.

3.340 IMR10 Control Policy (B_CR_BIMR10CP_0_0_0_MCHBAR) — Offset 69B8h

This register controls the access policy to the Read Access Policy BIMR10RAC, Write Access Policy BIMR10WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 69B8h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
60	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
59	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
58	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
55	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
52	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
50	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
47	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
44	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR10 Control Policy (IMR10_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR10 Control Policy (IMR10_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
41	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
40	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
39	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
36	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
33	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
31	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR10 Control Policy (IMR10_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR10 Control Policy (IMR10_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
28	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
25	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR10 Control Policy (IMR10_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
22	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
21	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
17	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR10 Control Policy (IMR10_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
14	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
12	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
9	1h RW	**IMR10 Control Policy (IMR10_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
6	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
3	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
2	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
1	1h RW	**IMR10 Control Policy (IMR10_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR10 Control Policy (IMR10_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR10RAC, BIMR10WAC and BIMR10CP registers based on the value from each agent's 6bit SAI field.

3.341 IMR10 Read Access Policy (B_CR_BIMR10RAC_0_0_0_MCHBAR) — Offset 69C0h

This register, along with IMR10BASE, IMR10MASK and IMR10WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR10. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 69C0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR10 Read Access Policy 63 (IMR10_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR10 Read Access Policy 62 (IMR10_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR10 Read Access Policy 61 (IMR10_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR10 Read Access Policy 60 (IMR10_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR10 Read Access Policy 59 (IMR10_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR10 Read Access Policy 58 (IMR10_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR10 Read Access Policy 57 (IMR10_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR10 Read Access Policy 56 (IMR10_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR10 Read Access Policy 55 (IMR10_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR10 Read Access Policy 54 (IMR10_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
53	0h RO	**IMR10 Read Access Policy 53 (IMR10_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR10 Read Access Policy 52 (IMR10_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR10 Read Access Policy 51 (IMR10_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR10 Read Access Policy 50 (IMR10_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR10 Read Access Policy 49 (IMR10_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR10 Read Access Policy 48 (IMR10_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR10 Read Access Policy 47 (IMR10_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR10 Read Access Policy 46 (IMR10_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR10 Read Access Policy 45 (IMR10_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR10 Read Access Policy 44 (IMR10_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR10 Read Access Policy 43 (IMR10_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR10 Read Access Policy 42 (IMR10_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR10 Read Access Policy 41 (IMR10_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR10 Read Access Policy 40 (IMR10_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR10 Read Access Policy 39 (IMR10_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR10 Read Access Policy 38 (IMR10_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR10 Read Access Policy 37 (IMR10_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR10 Read Access Policy 36 (IMR10_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR10 Read Access Policy 35 (IMR10_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
34	0h RW	**IMR10 Read Access Policy 34 (IMR10_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR10 Read Access Policy 33 (IMR10_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR10 Read Access Policy 32 (IMR10_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR10 Read Access Policy 31 (IMR10_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR10 Read Access Policy 30 (IMR10_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR10 Read Access Policy 29 (IMR10_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR10 Read Access Policy 28 (IMR10_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR10 Read Access Policy 27 (IMR10_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR10 Read Access Policy 26 (IMR10_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR10 Read Access Policy 25 (IMR10_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR10 Read Access Policy 24 (IMR10_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR10 Read Access Policy 23 (IMR10_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR10 Read Access Policy 22 (IMR10_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR10 Read Access Policy 21 (IMR10_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR10 Read Access Policy 20 (IMR10_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR10 Read Access Policy 19 (IMR10_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR10 Read Access Policy 18 (IMR10_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR10 Read Access Policy 17 (IMR10_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR10 Read Access Policy 16 (IMR10_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**IMR10 Read Access Policy 15 (IMR10_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR10 Read Access Policy 14 (IMR10_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR10 Read Access Policy 13 (IMR10_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR10 Read Access Policy 12 (IMR10_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR10 Read Access Policy 11 (IMR10_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR10 Read Access Policy 10 (IMR10_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR10 Read Access Policy 9 (IMR10_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR10 Read Access Policy 8 (IMR10_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR10 Read Access Policy 7 (IMR10_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR10 Read Access Policy 6 (IMR10_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR10 Read Access Policy 5 (IMR10_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR10 Read Access Policy 4 (IMR10_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR10 Read Access Policy 3 (IMR10_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR10 Read Access Policy 2 (IMR10_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR10 Read Access Policy 1 (IMR10_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR10 Read Access Policy 0 (IMR10_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR10 region, based on each agent's 6bit encoded SAI value.

3.342 IMR10 Write Access Policy (B_CR_BIMR10WAC_0_0_0_MCHBAR) — Offset 69C8h

This register, along with IMR10BASE, IMR10MASK and IMR10RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR10. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 69C8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR10 Write Access Policy 63 (IMR10_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR10 Write Access Policy 62 (IMR10_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR10 Write Access Policy 61 (IMR10_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR10 Write Access Policy 60 (IMR10_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR10 Write Access Policy 59 (IMR10_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR10 Write Access Policy 58 (IMR10_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR10 Write Access Policy 57 (IMR10_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR10 Write Access Policy 56 (IMR10_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR10 Write Access Policy 55 (IMR10_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR10 Write Access Policy 54 (IMR10_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR10 Write Access Policy 53 (IMR10_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR10 Write Access Policy 52 (IMR10_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR10 Write Access Policy 51 (IMR10_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
50	0h RW	**IMR10 Write Access Policy 50 (IMR10_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR10 Write Access Policy 49 (IMR10_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR10 Write Access Policy 48 (IMR10_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR10 Write Access Policy 47 (IMR10_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR10 Write Access Policy 46 (IMR10_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR10 Write Access Policy 45 (IMR10_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR10 Write Access Policy 44 (IMR10_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR10 Write Access Policy 43 (IMR10_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR10 Write Access Policy 42 (IMR10_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR10 Write Access Policy 41 (IMR10_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR10 Write Access Policy 40 (IMR10_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR10 Write Access Policy 39 (IMR10_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR10 Write Access Policy 38 (IMR10_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR10 Write Access Policy 37 (IMR10_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR10 Write Access Policy 36 (IMR10_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR10 Write Access Policy 35 (IMR10_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR10 Write Access Policy 34 (IMR10_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR10 Write Access Policy 33 (IMR10_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR10 Write Access Policy 32 (IMR10_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**IMR10 Write Access Policy 31 (IMR10_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR10 Write Access Policy 30 (IMR10_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR10 Write Access Policy 29 (IMR10_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR10 Write Access Policy 28 (IMR10_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR10 Write Access Policy 27 (IMR10_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR10 Write Access Policy 26 (IMR10_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR10 Write Access Policy 25 (IMR10_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR10 Write Access Policy 24 (IMR10_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR10 Write Access Policy 23 (IMR10_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR10 Write Access Policy 22 (IMR10_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR10 Write Access Policy 21 (IMR10_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR10 Write Access Policy 20 (IMR10_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR10 Write Access Policy 19 (IMR10_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR10 Write Access Policy 18 (IMR10_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR10 Write Access Policy 17 (IMR10_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR10 Write Access Policy 16 (IMR10_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR10 Write Access Policy 15 (IMR10_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR10 Write Access Policy 14 (IMR10_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR10 Write Access Policy 13 (IMR10_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**IMR10 Write Access Policy 12 (IMR10_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR10 Write Access Policy 11 (IMR10_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR10 Write Access Policy 10 (IMR10_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR10 Write Access Policy 9 (IMR10_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR10 Write Access Policy 8 (IMR10_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR10 Write Access Policy 7 (IMR10_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR10 Write Access Policy 6 (IMR10_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR10 Write Access Policy 5 (IMR10_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR10 Write Access Policy 4 (IMR10_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR10 Write Access Policy 3 (IMR10_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR10 Write Access Policy 2 (IMR10_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR10 Write Access Policy 1 (IMR10_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR10 Write Access Policy 0 (IMR10_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR10 region, based on each agent's 6bit encoded SAI value.

3.343 IMR11 Base (B_CR_BIMR11BASE_0_0_0_MCHBAR) — Offset 69D0h

This register, along with IMR11MASK, IMR11RAC, and IMR11WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR11RAC and IMR11WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 69D0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR11 Base (IMR11_BASE):** Specifies bits 38:10 of the start address of IMR11 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR11MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR11 defined region.

3.344 IMR11 Mask (B_CR_BIMR11MASK_0_0_0_MCHBAR) — Offset 69D4h

This register, along with IMR11BASE, IMR11RAC, and IMR11WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR11RAC and IMR11WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 69D4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved
28:0	0h RW	**Mask 0 IMR11 Mask (IMR11_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR11BASE[28:0] value. A match indicates that the incoming address falls within the IMR11 region.

3.345 IMR11 Control Policy (B_CR_BIMR11CP_0_0_0_MCHBAR) — Offset 69D8h

This register controls the access policy to the Read Access Policy BIMR11RAC, Write Access Policy BIMR11WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 69D8h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
60	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
59	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
58	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
55	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
52	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
50	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
47	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
44	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR11 Control Policy (IMR11_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR11 Control Policy (IMR11_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
41	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
40	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
39	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
36	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
33	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR11 Control Policy (IMR11_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR11 Control Policy (IMR11_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
28	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
25	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR11 Control Policy (IMR11_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
22	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
21	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
20	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
17	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR11 Control Policy (IMR11_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
14	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
9	1h RW	**IMR11 Control Policy (IMR11_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
6	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
3	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
2	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
1	1h RW	**IMR11 Control Policy (IMR11_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR11 Control Policy (IMR11_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR11RAC, BIMR11WAC and BIMR11CP registers based on the value from each agent's 6bit SAI field.

3.346 IMR11 Read Access Policy (B_CR_BIMR11RAC_0_0_0_MCHBAR) — Offset 69E0h

This register, along with IMR11BASE, IMR11MASK and IMR11WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR11. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 69E0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR11 Read Access Policy 63 (IMR11_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR11 Read Access Policy 62 (IMR11_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR11 Read Access Policy 61 (IMR11_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR11 Read Access Policy 60 (IMR11_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR11 Read Access Policy 59 (IMR11_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR11 Read Access Policy 58 (IMR11_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR11 Read Access Policy 57 (IMR11_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR11 Read Access Policy 56 (IMR11_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR11 Read Access Policy 55 (IMR11_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR11 Read Access Policy 54 (IMR11_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR11 Read Access Policy 53 (IMR11_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR11 Read Access Policy 52 (IMR11_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR11 Read Access Policy 51 (IMR11_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR11 Read Access Policy 50 (IMR11_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR11 Read Access Policy 49 (IMR11_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR11 Read Access Policy 48 (IMR11_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR11 Read Access Policy 47 (IMR11_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR11 Read Access Policy 46 (IMR11_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR11 Read Access Policy 45 (IMR11_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
44	0h RW	**IMR11 Read Access Policy 44 (IMR11_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR11 Read Access Policy 43 (IMR11_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR11 Read Access Policy 42 (IMR11_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR11 Read Access Policy 41 (IMR11_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR11 Read Access Policy 40 (IMR11_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR11 Read Access Policy 39 (IMR11_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR11 Read Access Policy 38 (IMR11_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR11 Read Access Policy 37 (IMR11_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR11 Read Access Policy 36 (IMR11_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR11 Read Access Policy 35 (IMR11_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR11 Read Access Policy 34 (IMR11_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR11 Read Access Policy 33 (IMR11_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR11 Read Access Policy 32 (IMR11_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR11 Read Access Policy 31 (IMR11_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR11 Read Access Policy 30 (IMR11_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR11 Read Access Policy 29 (IMR11_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR11 Read Access Policy 28 (IMR11_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR11 Read Access Policy 27 (IMR11_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR11 Read Access Policy 26 (IMR11_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**IMR11 Read Access Policy 25 (IMR11_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR11 Read Access Policy 24 (IMR11_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR11 Read Access Policy 23 (IMR11_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR11 Read Access Policy 22 (IMR11_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR11 Read Access Policy 21 (IMR11_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR11 Read Access Policy 20 (IMR11_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR11 Read Access Policy 19 (IMR11_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR11 Read Access Policy 18 (IMR11_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR11 Read Access Policy 17 (IMR11_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR11 Read Access Policy 16 (IMR11_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR11 Read Access Policy 15 (IMR11_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR11 Read Access Policy 14 (IMR11_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR11 Read Access Policy 13 (IMR11_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR11 Read Access Policy 12 (IMR11_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR11 Read Access Policy 11 (IMR11_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR11 Read Access Policy 10 (IMR11_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR11 Read Access Policy 9 (IMR11_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR11 Read Access Policy 8 (IMR11_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR11 Read Access Policy 7 (IMR11_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**IMR11 Read Access Policy 6 (IMR11_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR11 Read Access Policy 5 (IMR11_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR11 Read Access Policy 4 (IMR11_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR11 Read Access Policy 3 (IMR11_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR11 Read Access Policy 2 (IMR11_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR11 Read Access Policy 1 (IMR11_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR11 Read Access Policy 0 (IMR11_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR11 region, based on each agent's 6bit encoded SAI value.

3.347 IMR11 Write Access Policy (B_CR_BIMR11WAC_0_0_0_MCHBAR) — Offset 69E8h

This register, along with IMR11BASE, IMR11MASK and IMR11RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR11. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 69E8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR11 Write Access Policy 63 (IMR11_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR11 Write Access Policy 62 (IMR11_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR11 Write Access Policy 61 (IMR11_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR11 Write Access Policy 60 (IMR11_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR11 Write Access Policy 59 (IMR11_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
58	0h RO	**IMR11 Write Access Policy 58 (IMR11_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR11 Write Access Policy 57 (IMR11_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR11 Write Access Policy 56 (IMR11_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR11 Write Access Policy 55 (IMR11_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR11 Write Access Policy 54 (IMR11_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR11 Write Access Policy 53 (IMR11_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR11 Write Access Policy 52 (IMR11_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR11 Write Access Policy 51 (IMR11_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR11 Write Access Policy 50 (IMR11_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR11 Write Access Policy 49 (IMR11_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR11 Write Access Policy 48 (IMR11_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR11 Write Access Policy 47 (IMR11_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR11 Write Access Policy 46 (IMR11_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR11 Write Access Policy 45 (IMR11_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR11 Write Access Policy 44 (IMR11_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR11 Write Access Policy 43 (IMR11_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR11 Write Access Policy 42 (IMR11_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR11 Write Access Policy 41 (IMR11_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR11 Write Access Policy 40 (IMR11_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
39	0h RO	**IMR11 Write Access Policy 39 (IMR11_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR11 Write Access Policy 38 (IMR11_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR11 Write Access Policy 37 (IMR11_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR11 Write Access Policy 36 (IMR11_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR11 Write Access Policy 35 (IMR11_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR11 Write Access Policy 34 (IMR11_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR11 Write Access Policy 33 (IMR11_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR11 Write Access Policy 32 (IMR11_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR11 Write Access Policy 31 (IMR11_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR11 Write Access Policy 30 (IMR11_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR11 Write Access Policy 29 (IMR11_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR11 Write Access Policy 28 (IMR11_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR11 Write Access Policy 27 (IMR11_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR11 Write Access Policy 26 (IMR11_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR11 Write Access Policy 25 (IMR11_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR11 Write Access Policy 24 (IMR11_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR11 Write Access Policy 23 (IMR11_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR11 Write Access Policy 22 (IMR11_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR11 Write Access Policy 21 (IMR11_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**IMR11 Write Access Policy 20 (IMR11_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR11 Write Access Policy 19 (IMR11_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR11 Write Access Policy 18 (IMR11_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR11 Write Access Policy 17 (IMR11_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR11 Write Access Policy 16 (IMR11_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR11 Write Access Policy 15 (IMR11_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR11 Write Access Policy 14 (IMR11_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR11 Write Access Policy 13 (IMR11_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR11 Write Access Policy 12 (IMR11_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR11 Write Access Policy 11 (IMR11_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR11 Write Access Policy 10 (IMR11_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR11 Write Access Policy 9 (IMR11_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR11 Write Access Policy 8 (IMR11_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR11 Write Access Policy 7 (IMR11_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR11 Write Access Policy 6 (IMR11_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR11 Write Access Policy 5 (IMR11_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR11 Write Access Policy 4 (IMR11_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR11 Write Access Policy 3 (IMR11_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR11 Write Access Policy 2 (IMR11_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**IMR11 Write Access Policy 1 (IMR11_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR11 Write Access Policy 0 (IMR11_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR11 region, based on each agent's 6bit encoded SAI value.

3.348 IMR12 Base (B_CR_BIMR12BASE_0_0_0_MCHBAR) — Offset 69F0h

This register, along with IMR12MASK, IMR12RAC, and IMR12WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR12RAC and IMR12WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 69F0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR12 Base (IMR12_BASE):** Specifies bits 38:10 of the start address of IMR12 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR12MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR12 defined region.

3.349 IMR12 Mask (B_CR_BIMR12MASK_0_0_0_MCHBAR) — Offset 69F4h

This register, along with IMR12BASE, IMR12RAC, and IMR12WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR12RAC and IMR12WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 69F4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved
28:0	0h RW	**Mask 0 IMR12 Mask (IMR12_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR12BASE[28:0] value. A match indicates that the incoming address falls within the IMR12 region.

3.350 IMR12 Control Policy (B_CR_BIMR12CP_0_0_0_MCHBAR) — Offset 69F8h

This register controls the access policy to the Read Access Policy BIMR12RAC, Write Access Policy BIMR12WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 69F8h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
60	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
59	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
58	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
55	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
52	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
50	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
47	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
44	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR12 Control Policy (IMR12_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR12 Control Policy (IMR12_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
41	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
40	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
39	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
36	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
33	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
31	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR12 Control Policy (IMR12_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR12 Control Policy (IMR12_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
28	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
25	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR12 Control Policy (IMR12_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
22	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
20	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
17	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR12 Control Policy (IMR12_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
14	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
12	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
9	1h RW	**IMR12 Control Policy (IMR12_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
6	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
3	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
1	1h RW	**IMR12 Control Policy (IMR12_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR12 Control Policy (IMR12_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR12RAC, BIMR12WAC and BIMR12CP registers based on the value from each agent's 6bit SAI field.

3.351 IMR12 Read Access Policy (B_CR_BIMR12RAC_0_0_0_MCHBAR) — Offset 6A00h

This register, along with IMR12BASE, IMR12MASK and IMR12WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR12. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6A00h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR12 Read Access Policy 63 (IMR12_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR12 Read Access Policy 62 (IMR12_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR12 Read Access Policy 61 (IMR12_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR12 Read Access Policy 60 (IMR12_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR12 Read Access Policy 59 (IMR12_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR12 Read Access Policy 58 (IMR12_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR12 Read Access Policy 57 (IMR12_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR12 Read Access Policy 56 (IMR12_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR12 Read Access Policy 55 (IMR12_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
54	0h RW	**IMR12 Read Access Policy 54 (IMR12_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR12 Read Access Policy 53 (IMR12_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR12 Read Access Policy 52 (IMR12_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR12 Read Access Policy 51 (IMR12_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR12 Read Access Policy 50 (IMR12_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR12 Read Access Policy 49 (IMR12_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR12 Read Access Policy 48 (IMR12_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR12 Read Access Policy 47 (IMR12_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR12 Read Access Policy 46 (IMR12_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR12 Read Access Policy 45 (IMR12_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR12 Read Access Policy 44 (IMR12_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR12 Read Access Policy 43 (IMR12_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR12 Read Access Policy 42 (IMR12_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR12 Read Access Policy 41 (IMR12_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR12 Read Access Policy 40 (IMR12_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR12 Read Access Policy 39 (IMR12_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR12 Read Access Policy 38 (IMR12_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR12 Read Access Policy 37 (IMR12_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR12 Read Access Policy 36 (IMR12_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
35	0h RO	**IMR12 Read Access Policy 35 (IMR12_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR12 Read Access Policy 34 (IMR12_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR12 Read Access Policy 33 (IMR12_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR12 Read Access Policy 32 (IMR12_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR12 Read Access Policy 31 (IMR12_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR12 Read Access Policy 30 (IMR12_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR12 Read Access Policy 29 (IMR12_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR12 Read Access Policy 28 (IMR12_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR12 Read Access Policy 27 (IMR12_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR12 Read Access Policy 26 (IMR12_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR12 Read Access Policy 25 (IMR12_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR12 Read Access Policy 24 (IMR12_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR12 Read Access Policy 23 (IMR12_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR12 Read Access Policy 22 (IMR12_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR12 Read Access Policy 21 (IMR12_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR12 Read Access Policy 20 (IMR12_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR12 Read Access Policy 19 (IMR12_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR12 Read Access Policy 18 (IMR12_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR12 Read Access Policy 17 (IMR12_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW	**IMR12 Read Access Policy 16 (IMR12_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR12 Read Access Policy 15 (IMR12_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR12 Read Access Policy 14 (IMR12_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR12 Read Access Policy 13 (IMR12_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR12 Read Access Policy 12 (IMR12_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR12 Read Access Policy 11 (IMR12_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR12 Read Access Policy 10 (IMR12_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR12 Read Access Policy 9 (IMR12_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR12 Read Access Policy 8 (IMR12_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR12 Read Access Policy 7 (IMR12_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR12 Read Access Policy 6 (IMR12_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR12 Read Access Policy 5 (IMR12_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR12 Read Access Policy 4 (IMR12_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR12 Read Access Policy 3 (IMR12_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR12 Read Access Policy 2 (IMR12_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR12 Read Access Policy 1 (IMR12_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR12 Read Access Policy 0 (IMR12_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR12 region, based on each agent's 6bit encoded SAI value.

3.352 IMR12 Write Access Policy (B_CR_BIMR12WAC_0_0_0_MCHBAR) — Offset 6A08h

This register, along with IMR12BASE, IMR12MASK and IMR12RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR12. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6A08h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR12 Write Access Policy 63 (IMR12_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR12 Write Access Policy 62 (IMR12_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR12 Write Access Policy 61 (IMR12_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR12 Write Access Policy 60 (IMR12_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR12 Write Access Policy 59 (IMR12_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR12 Write Access Policy 58 (IMR12_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR12 Write Access Policy 57 (IMR12_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR12 Write Access Policy 56 (IMR12_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR12 Write Access Policy 55 (IMR12_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR12 Write Access Policy 54 (IMR12_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR12 Write Access Policy 53 (IMR12_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR12 Write Access Policy 52 (IMR12_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR12 Write Access Policy 51 (IMR12_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
50	0h RW	**IMR12 Write Access Policy 50 (IMR12_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR12 Write Access Policy 49 (IMR12_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR12 Write Access Policy 48 (IMR12_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR12 Write Access Policy 47 (IMR12_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR12 Write Access Policy 46 (IMR12_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR12 Write Access Policy 45 (IMR12_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR12 Write Access Policy 44 (IMR12_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR12 Write Access Policy 43 (IMR12_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR12 Write Access Policy 42 (IMR12_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR12 Write Access Policy 41 (IMR12_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR12 Write Access Policy 40 (IMR12_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR12 Write Access Policy 39 (IMR12_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR12 Write Access Policy 38 (IMR12_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR12 Write Access Policy 37 (IMR12_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR12 Write Access Policy 36 (IMR12_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR12 Write Access Policy 35 (IMR12_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR12 Write Access Policy 34 (IMR12_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR12 Write Access Policy 33 (IMR12_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR12 Write Access Policy 32 (IMR12_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**IMR12 Write Access Policy 31 (IMR12_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR12 Write Access Policy 30 (IMR12_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR12 Write Access Policy 29 (IMR12_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR12 Write Access Policy 28 (IMR12_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR12 Write Access Policy 27 (IMR12_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR12 Write Access Policy 26 (IMR12_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR12 Write Access Policy 25 (IMR12_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR12 Write Access Policy 24 (IMR12_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR12 Write Access Policy 23 (IMR12_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR12 Write Access Policy 22 (IMR12_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR12 Write Access Policy 21 (IMR12_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR12 Write Access Policy 20 (IMR12_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR12 Write Access Policy 19 (IMR12_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR12 Write Access Policy 18 (IMR12_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR12 Write Access Policy 17 (IMR12_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR12 Write Access Policy 16 (IMR12_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR12 Write Access Policy 15 (IMR12_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR12 Write Access Policy 14 (IMR12_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR12 Write Access Policy 13 (IMR12_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**IMR12 Write Access Policy 12 (IMR12_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR12 Write Access Policy 11 (IMR12_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR12 Write Access Policy 10 (IMR12_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR12 Write Access Policy 9 (IMR12_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR12 Write Access Policy 8 (IMR12_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR12 Write Access Policy 7 (IMR12_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR12 Write Access Policy 6 (IMR12_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR12 Write Access Policy 5 (IMR12_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR12 Write Access Policy 4 (IMR12_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR12 Write Access Policy 3 (IMR12_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR12 Write Access Policy 2 (IMR12_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR12 Write Access Policy 1 (IMR12_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR12 Write Access Policy 0 (IMR12_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR12 region, based on each agent's 6bit encoded SAI value.

3.353 IMR13 Base (B_CR_BIMR13BASE_0_0_0_MCHBAR) — Offset 6A10h

This register, along with IMR13MASK, IMR13RAC, and IMR13WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR13RAC and IMR13WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6A10h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR13 Base (IMR13_BASE):** Specifies bits 38:10 of the start address of IMR13 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR13MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR13 defined region.

3.354 IMR13 Mask (B_CR_BIMR13MASK_0_0_0_MCHBAR) — Offset 6A14h

This register, along with IMR13BASE, IMR13RAC, and IMR13WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR13RAC and IMR13WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6A14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved
28:0	0h RW	**Mask 0 IMR13 Mask (IMR13_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR13BASE[28:0] value. A match indicates that the incoming address falls within the IMR13 region.

3.355 IMR13 Control Policy (B_CR_BIMR13CP_0_0_0_MCHBAR) — Offset 6A18h

This register controls the access policy to the Read Access Policy BIMR13RAC, Write Access Policy BIMR13WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6A18h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
60	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
59	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
58	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
55	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
52	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
50	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
47	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
44	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR13 Control Policy (IMR13_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR13 Control Policy (IMR13_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
41	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
40	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
39	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
36	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
33	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR13 Control Policy (IMR13_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR13 Control Policy (IMR13_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
28	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
25	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR13 Control Policy (IMR13_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
22	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
21	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
20	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
17	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR13 Control Policy (IMR13_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
14	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
9	1h RW	**IMR13 Control Policy (IMR13_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
6	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
3	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
2	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
1	1h RW	**IMR13 Control Policy (IMR13_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR13 Control Policy (IMR13_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR13RAC, BIMR13WAC and BIMR13CP registers based on the value from each agent's 6bit SAI field.

3.356 IMR13 Read Access Policy (B_CR_BIMR13RAC_0_0_0_MCHBAR) — Offset 6A20h

This register, along with IMR13BASE, IMR13MASK and IMR13WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR13. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6A20h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR13 Read Access Policy 63 (IMR13_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR13 Read Access Policy 62 (IMR13_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR13 Read Access Policy 61 (IMR13_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR13 Read Access Policy 60 (IMR13_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR13 Read Access Policy 59 (IMR13_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR13 Read Access Policy 58 (IMR13_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR13 Read Access Policy 57 (IMR13_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR13 Read Access Policy 56 (IMR13_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR13 Read Access Policy 55 (IMR13_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR13 Read Access Policy 54 (IMR13_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR13 Read Access Policy 53 (IMR13_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR13 Read Access Policy 52 (IMR13_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR13 Read Access Policy 51 (IMR13_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR13 Read Access Policy 50 (IMR13_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR13 Read Access Policy 49 (IMR13_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR13 Read Access Policy 48 (IMR13_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR13 Read Access Policy 47 (IMR13_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR13 Read Access Policy 46 (IMR13_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR13 Read Access Policy 45 (IMR13_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
44	0h RW	**IMR13 Read Access Policy 44 (IMR13_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR13 Read Access Policy 43 (IMR13_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR13 Read Access Policy 42 (IMR13_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR13 Read Access Policy 41 (IMR13_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR13 Read Access Policy 40 (IMR13_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR13 Read Access Policy 39 (IMR13_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR13 Read Access Policy 38 (IMR13_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR13 Read Access Policy 37 (IMR13_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR13 Read Access Policy 36 (IMR13_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR13 Read Access Policy 35 (IMR13_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR13 Read Access Policy 34 (IMR13_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR13 Read Access Policy 33 (IMR13_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR13 Read Access Policy 32 (IMR13_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR13 Read Access Policy 31 (IMR13_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR13 Read Access Policy 30 (IMR13_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR13 Read Access Policy 29 (IMR13_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR13 Read Access Policy 28 (IMR13_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR13 Read Access Policy 27 (IMR13_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR13 Read Access Policy 26 (IMR13_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**IMR13 Read Access Policy 25 (IMR13_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR13 Read Access Policy 24 (IMR13_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR13 Read Access Policy 23 (IMR13_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR13 Read Access Policy 22 (IMR13_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR13 Read Access Policy 21 (IMR13_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR13 Read Access Policy 20 (IMR13_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR13 Read Access Policy 19 (IMR13_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR13 Read Access Policy 18 (IMR13_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR13 Read Access Policy 17 (IMR13_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR13 Read Access Policy 16 (IMR13_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR13 Read Access Policy 15 (IMR13_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR13 Read Access Policy 14 (IMR13_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR13 Read Access Policy 13 (IMR13_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR13 Read Access Policy 12 (IMR13_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR13 Read Access Policy 11 (IMR13_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR13 Read Access Policy 10 (IMR13_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR13 Read Access Policy 9 (IMR13_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR13 Read Access Policy 8 (IMR13_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR13 Read Access Policy 7 (IMR13_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**IMR13 Read Access Policy 6 (IMR13_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR13 Read Access Policy 5 (IMR13_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR13 Read Access Policy 4 (IMR13_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR13 Read Access Policy 3 (IMR13_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR13 Read Access Policy 2 (IMR13_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR13 Read Access Policy 1 (IMR13_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR13 Read Access Policy 0 (IMR13_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR13 region, based on each agent's 6bit encoded SAI value.

3.357 IMR13 Write Access Policy (B_CR_BIMR13WAC_0_0_0_MCHBAR) — Offset 6A28h

This register, along with IMR13BASE, IMR13MASK and IMR13RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR13. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6A28h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR13 Write Access Policy 63 (IMR13_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR13 Write Access Policy 62 (IMR13_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR13 Write Access Policy 61 (IMR13_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR13 Write Access Policy 60 (IMR13_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR13 Write Access Policy 59 (IMR13_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
58	0h RO	**IMR13 Write Access Policy 58 (IMR13_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR13 Write Access Policy 57 (IMR13_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR13 Write Access Policy 56 (IMR13_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR13 Write Access Policy 55 (IMR13_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR13 Write Access Policy 54 (IMR13_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR13 Write Access Policy 53 (IMR13_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR13 Write Access Policy 52 (IMR13_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR13 Write Access Policy 51 (IMR13_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR13 Write Access Policy 50 (IMR13_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR13 Write Access Policy 49 (IMR13_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR13 Write Access Policy 48 (IMR13_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR13 Write Access Policy 47 (IMR13_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR13 Write Access Policy 46 (IMR13_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR13 Write Access Policy 45 (IMR13_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR13 Write Access Policy 44 (IMR13_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR13 Write Access Policy 43 (IMR13_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR13 Write Access Policy 42 (IMR13_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR13 Write Access Policy 41 (IMR13_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR13 Write Access Policy 40 (IMR13_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
39	0h RO	**IMR13 Write Access Policy 39 (IMR13_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR13 Write Access Policy 38 (IMR13_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR13 Write Access Policy 37 (IMR13_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR13 Write Access Policy 36 (IMR13_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR13 Write Access Policy 35 (IMR13_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR13 Write Access Policy 34 (IMR13_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR13 Write Access Policy 33 (IMR13_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR13 Write Access Policy 32 (IMR13_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR13 Write Access Policy 31 (IMR13_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR13 Write Access Policy 30 (IMR13_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR13 Write Access Policy 29 (IMR13_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR13 Write Access Policy 28 (IMR13_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR13 Write Access Policy 27 (IMR13_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR13 Write Access Policy 26 (IMR13_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR13 Write Access Policy 25 (IMR13_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR13 Write Access Policy 24 (IMR13_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR13 Write Access Policy 23 (IMR13_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR13 Write Access Policy 22 (IMR13_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR13 Write Access Policy 21 (IMR13_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**IMR13 Write Access Policy 20 (IMR13_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR13 Write Access Policy 19 (IMR13_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR13 Write Access Policy 18 (IMR13_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR13 Write Access Policy 17 (IMR13_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR13 Write Access Policy 16 (IMR13_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR13 Write Access Policy 15 (IMR13_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR13 Write Access Policy 14 (IMR13_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR13 Write Access Policy 13 (IMR13_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR13 Write Access Policy 12 (IMR13_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR13 Write Access Policy 11 (IMR13_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR13 Write Access Policy 10 (IMR13_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR13 Write Access Policy 9 (IMR13_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR13 Write Access Policy 8 (IMR13_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR13 Write Access Policy 7 (IMR13_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR13 Write Access Policy 6 (IMR13_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR13 Write Access Policy 5 (IMR13_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR13 Write Access Policy 4 (IMR13_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR13 Write Access Policy 3 (IMR13_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR13 Write Access Policy 2 (IMR13_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**IMR13 Write Access Policy 1 (IMR13_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR13 Write Access Policy 0 (IMR13_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR13 region, based on each agent's 6bit encoded SAI value.

3.358 IMR14 Base (B_CR_BIMR14BASE_0_0_0_MCHBAR) — Offset 6A30h

This register, along with IMR14MASK, IMR14RAC, and IMR14WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR14RAC and IMR14WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6A30h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR14 Base (IMR14_BASE):** Specifies bits 38:10 of the start address of IMR14 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR14MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR14 defined region.

3.359 IMR14 Mask (B_CR_BIMR14MASK_0_0_0_MCHBAR) — Offset 6A34h

This register, along with IMR14BASE, IMR14RAC, and IMR14WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR14RAC and IMR14WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6A34h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved
28:0	0h RW	**Mask 0 IMR14 Mask (IMR14_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR14BASE[28:0] value. A match indicates that the incoming address falls within the IMR14 region.

3.360 IMR14 Control Policy (B_CR_BIMR14CP_0_0_0_MCHBAR) — Offset 6A38h

This register controls the access policy to the Read Access Policy BIMR14RAC, Write Access Policy BIMR14WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6A38h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
60	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
59	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
58	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
55	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
52	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
50	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
47	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
44	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR14 Control Policy (IMR14_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR14 Control Policy (IMR14_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
41	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
40	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
39	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
36	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
33	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
31	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR14 Control Policy (IMR14_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR14 Control Policy (IMR14_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
28	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
25	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR14 Control Policy (IMR14_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
22	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
21	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
17	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR14 Control Policy (IMR14_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
14	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
12	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
9	1h RW	**IMR14 Control Policy (IMR14_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
6	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
3	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
2	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
1	1h RW	**IMR14 Control Policy (IMR14_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR14 Control Policy (IMR14_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR14RAC, BIMR14WAC and BIMR14CP registers based on the value from each agent's 6bit SAI field.

3.361 IMR14 Read Access Policy (B_CR_BIMR14RAC_0_0_0_MCHBAR) — Offset 6A40h

This register, along with IMR14BASE, IMR14MASK and IMR14WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR14. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6A40h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR14 Read Access Policy 63 (IMR14_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR14 Read Access Policy 62 (IMR14_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR14 Read Access Policy 61 (IMR14_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR14 Read Access Policy 60 (IMR14_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR14 Read Access Policy 59 (IMR14_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR14 Read Access Policy 58 (IMR14_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR14 Read Access Policy 57 (IMR14_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR14 Read Access Policy 56 (IMR14_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR14 Read Access Policy 55 (IMR14_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR14 Read Access Policy 54 (IMR14_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
53	0h RO	**IMR14 Read Access Policy 53 (IMR14_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR14 Read Access Policy 52 (IMR14_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR14 Read Access Policy 51 (IMR14_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR14 Read Access Policy 50 (IMR14_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR14 Read Access Policy 49 (IMR14_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR14 Read Access Policy 48 (IMR14_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR14 Read Access Policy 47 (IMR14_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR14 Read Access Policy 46 (IMR14_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR14 Read Access Policy 45 (IMR14_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR14 Read Access Policy 44 (IMR14_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR14 Read Access Policy 43 (IMR14_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR14 Read Access Policy 42 (IMR14_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR14 Read Access Policy 41 (IMR14_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR14 Read Access Policy 40 (IMR14_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR14 Read Access Policy 39 (IMR14_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR14 Read Access Policy 38 (IMR14_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR14 Read Access Policy 37 (IMR14_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR14 Read Access Policy 36 (IMR14_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR14 Read Access Policy 35 (IMR14_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
34	0h RW	**IMR14 Read Access Policy 34 (IMR14_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR14 Read Access Policy 33 (IMR14_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR14 Read Access Policy 32 (IMR14_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR14 Read Access Policy 31 (IMR14_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR14 Read Access Policy 30 (IMR14_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR14 Read Access Policy 29 (IMR14_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR14 Read Access Policy 28 (IMR14_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR14 Read Access Policy 27 (IMR14_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR14 Read Access Policy 26 (IMR14_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR14 Read Access Policy 25 (IMR14_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR14 Read Access Policy 24 (IMR14_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR14 Read Access Policy 23 (IMR14_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR14 Read Access Policy 22 (IMR14_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR14 Read Access Policy 21 (IMR14_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR14 Read Access Policy 20 (IMR14_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR14 Read Access Policy 19 (IMR14_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR14 Read Access Policy 18 (IMR14_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR14 Read Access Policy 17 (IMR14_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR14 Read Access Policy 16 (IMR14_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**IMR14 Read Access Policy 15 (IMR14_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR14 Read Access Policy 14 (IMR14_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR14 Read Access Policy 13 (IMR14_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR14 Read Access Policy 12 (IMR14_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR14 Read Access Policy 11 (IMR14_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR14 Read Access Policy 10 (IMR14_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR14 Read Access Policy 9 (IMR14_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR14 Read Access Policy 8 (IMR14_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR14 Read Access Policy 7 (IMR14_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR14 Read Access Policy 6 (IMR14_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR14 Read Access Policy 5 (IMR14_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR14 Read Access Policy 4 (IMR14_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR14 Read Access Policy 3 (IMR14_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR14 Read Access Policy 2 (IMR14_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR14 Read Access Policy 1 (IMR14_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR14 Read Access Policy 0 (IMR14_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR14 region, based on each agent's 6bit encoded SAI value.

3.362 IMR14 Write Access Policy (B_CR_BIMR14WAC_0_0_0_MCHBAR) — Offset 6A48h

This register, along with IMR14BASE, IMR14MASK and IMR14RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR14. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6A48h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR14 Write Access Policy 63 (IMR14_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR14 Write Access Policy 62 (IMR14_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR14 Write Access Policy 61 (IMR14_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR14 Write Access Policy 60 (IMR14_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR14 Write Access Policy 59 (IMR14_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR14 Write Access Policy 58 (IMR14_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR14 Write Access Policy 57 (IMR14_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR14 Write Access Policy 56 (IMR14_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR14 Write Access Policy 55 (IMR14_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR14 Write Access Policy 54 (IMR14_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR14 Write Access Policy 53 (IMR14_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR14 Write Access Policy 52 (IMR14_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR14 Write Access Policy 51 (IMR14_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
50	0h RW	**IMR14 Write Access Policy 50 (IMR14_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR14 Write Access Policy 49 (IMR14_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR14 Write Access Policy 48 (IMR14_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR14 Write Access Policy 47 (IMR14_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR14 Write Access Policy 46 (IMR14_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR14 Write Access Policy 45 (IMR14_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR14 Write Access Policy 44 (IMR14_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR14 Write Access Policy 43 (IMR14_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR14 Write Access Policy 42 (IMR14_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR14 Write Access Policy 41 (IMR14_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR14 Write Access Policy 40 (IMR14_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR14 Write Access Policy 39 (IMR14_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR14 Write Access Policy 38 (IMR14_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR14 Write Access Policy 37 (IMR14_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR14 Write Access Policy 36 (IMR14_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR14 Write Access Policy 35 (IMR14_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR14 Write Access Policy 34 (IMR14_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR14 Write Access Policy 33 (IMR14_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR14 Write Access Policy 32 (IMR14_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**IMR14 Write Access Policy 31 (IMR14_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR14 Write Access Policy 30 (IMR14_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR14 Write Access Policy 29 (IMR14_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR14 Write Access Policy 28 (IMR14_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR14 Write Access Policy 27 (IMR14_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR14 Write Access Policy 26 (IMR14_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR14 Write Access Policy 25 (IMR14_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR14 Write Access Policy 24 (IMR14_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR14 Write Access Policy 23 (IMR14_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR14 Write Access Policy 22 (IMR14_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR14 Write Access Policy 21 (IMR14_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR14 Write Access Policy 20 (IMR14_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR14 Write Access Policy 19 (IMR14_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR14 Write Access Policy 18 (IMR14_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR14 Write Access Policy 17 (IMR14_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR14 Write Access Policy 16 (IMR14_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR14 Write Access Policy 15 (IMR14_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR14 Write Access Policy 14 (IMR14_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR14 Write Access Policy 13 (IMR14_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**IMR14 Write Access Policy 12 (IMR14_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR14 Write Access Policy 11 (IMR14_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR14 Write Access Policy 10 (IMR14_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR14 Write Access Policy 9 (IMR14_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR14 Write Access Policy 8 (IMR14_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR14 Write Access Policy 7 (IMR14_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR14 Write Access Policy 6 (IMR14_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR14 Write Access Policy 5 (IMR14_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR14 Write Access Policy 4 (IMR14_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR14 Write Access Policy 3 (IMR14_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR14 Write Access Policy 2 (IMR14_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR14 Write Access Policy 1 (IMR14_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR14 Write Access Policy 0 (IMR14_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR14 region, based on each agent's 6bit encoded SAI value.

3.363 IMR15 Base (B_CR_BIMR15BASE_0_0_0_MCHBAR) — Offset 6A50h

This register, along with IMR15MASK, IMR15RAC, and IMR15WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR15RAC and IMR15WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6A50h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR15 Base (IMR15_BASE):** Specifies bits 38:10 of the start address of IMR15 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR15MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR15 defined region.

3.364 IMR15 Mask (B_CR_BIMR15MASK_0_0_0_MCHBAR) — Offset 6A54h

This register, along with IMR15BASE, IMR15RAC, and IMR15WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR15RAC and IMR15WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6A54h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved
28:0	0h RW	**Mask 0 IMR15 Mask (IMR15_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR15BASE[28:0] value. A match indicates that the incoming address falls within the IMR15 region.

3.365 IMR15 Control Policy (B_CR_BIMR15CP_0_0_0_MCHBAR) — Offset 6A58h

This register controls the access policy to the Read Access Policy BIMR15RAC, Write Access Policy BIMR15WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6A58h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
60	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
59	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
58	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
55	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
52	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
50	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
47	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
44	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR15 Control Policy (IMR15_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR15 Control Policy (IMR15_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
41	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
40	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
39	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
36	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
33	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR15 Control Policy (IMR15_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR15 Control Policy (IMR15_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
28	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
25	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR15 Control Policy (IMR15_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
22	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
21	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
20	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
17	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR15 Control Policy (IMR15_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
14	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
9	1h RW	**IMR15 Control Policy (IMR15_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
6	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
3	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
2	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
1	1h RW	**IMR15 Control Policy (IMR15_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR15 Control Policy (IMR15_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR15RAC, BIMR15WAC and BIMR15CP registers based on the value from each agent's 6bit SAI field.

3.366 IMR15 Read Access Policy (B_CR_BIMR15RAC_0_0_0_MCHBAR) — Offset 6A60h

This register, along with IMR15BASE, IMR15MASK and IMR15WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR15. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6A60h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR15 Read Access Policy 63 (IMR15_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR15 Read Access Policy 62 (IMR15_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR15 Read Access Policy 61 (IMR15_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR15 Read Access Policy 60 (IMR15_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR15 Read Access Policy 59 (IMR15_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR15 Read Access Policy 58 (IMR15_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR15 Read Access Policy 57 (IMR15_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR15 Read Access Policy 56 (IMR15_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR15 Read Access Policy 55 (IMR15_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR15 Read Access Policy 54 (IMR15_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR15 Read Access Policy 53 (IMR15_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR15 Read Access Policy 52 (IMR15_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR15 Read Access Policy 51 (IMR15_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR15 Read Access Policy 50 (IMR15_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR15 Read Access Policy 49 (IMR15_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR15 Read Access Policy 48 (IMR15_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR15 Read Access Policy 47 (IMR15_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR15 Read Access Policy 46 (IMR15_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR15 Read Access Policy 45 (IMR15_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
44	0h RW	**IMR15 Read Access Policy 44 (IMR15_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR15 Read Access Policy 43 (IMR15_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR15 Read Access Policy 42 (IMR15_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR15 Read Access Policy 41 (IMR15_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR15 Read Access Policy 40 (IMR15_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR15 Read Access Policy 39 (IMR15_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR15 Read Access Policy 38 (IMR15_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR15 Read Access Policy 37 (IMR15_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR15 Read Access Policy 36 (IMR15_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR15 Read Access Policy 35 (IMR15_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR15 Read Access Policy 34 (IMR15_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR15 Read Access Policy 33 (IMR15_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR15 Read Access Policy 32 (IMR15_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR15 Read Access Policy 31 (IMR15_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR15 Read Access Policy 30 (IMR15_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR15 Read Access Policy 29 (IMR15_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR15 Read Access Policy 28 (IMR15_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR15 Read Access Policy 27 (IMR15_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR15 Read Access Policy 26 (IMR15_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**IMR15 Read Access Policy 25 (IMR15_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR15 Read Access Policy 24 (IMR15_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR15 Read Access Policy 23 (IMR15_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR15 Read Access Policy 22 (IMR15_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR15 Read Access Policy 21 (IMR15_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR15 Read Access Policy 20 (IMR15_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR15 Read Access Policy 19 (IMR15_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR15 Read Access Policy 18 (IMR15_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR15 Read Access Policy 17 (IMR15_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR15 Read Access Policy 16 (IMR15_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR15 Read Access Policy 15 (IMR15_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR15 Read Access Policy 14 (IMR15_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR15 Read Access Policy 13 (IMR15_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR15 Read Access Policy 12 (IMR15_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR15 Read Access Policy 11 (IMR15_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR15 Read Access Policy 10 (IMR15_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR15 Read Access Policy 9 (IMR15_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR15 Read Access Policy 8 (IMR15_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR15 Read Access Policy 7 (IMR15_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**IMR15 Read Access Policy 6 (IMR15_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR15 Read Access Policy 5 (IMR15_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR15 Read Access Policy 4 (IMR15_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR15 Read Access Policy 3 (IMR15_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR15 Read Access Policy 2 (IMR15_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR15 Read Access Policy 1 (IMR15_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR15 Read Access Policy 0 (IMR15_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR15 region, based on each agent's 6bit encoded SAI value.

3.367 IMR15 Write Access Policy (B_CR_BIMR15WAC_0_0_0_MCHBAR) — Offset 6A68h

This register, along with IMR15BASE, IMR15MASK and IMR15RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR15. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6A68h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR15 Write Access Policy 63 (IMR15_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR15 Write Access Policy 62 (IMR15_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR15 Write Access Policy 61 (IMR15_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR15 Write Access Policy 60 (IMR15_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR15 Write Access Policy 59 (IMR15_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
58	0h RO	**IMR15 Write Access Policy 58 (IMR15_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR15 Write Access Policy 57 (IMR15_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR15 Write Access Policy 56 (IMR15_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR15 Write Access Policy 55 (IMR15_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR15 Write Access Policy 54 (IMR15_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR15 Write Access Policy 53 (IMR15_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR15 Write Access Policy 52 (IMR15_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR15 Write Access Policy 51 (IMR15_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR15 Write Access Policy 50 (IMR15_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR15 Write Access Policy 49 (IMR15_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR15 Write Access Policy 48 (IMR15_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR15 Write Access Policy 47 (IMR15_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR15 Write Access Policy 46 (IMR15_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR15 Write Access Policy 45 (IMR15_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR15 Write Access Policy 44 (IMR15_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR15 Write Access Policy 43 (IMR15_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR15 Write Access Policy 42 (IMR15_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR15 Write Access Policy 41 (IMR15_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR15 Write Access Policy 40 (IMR15_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
39	0h RO	**IMR15 Write Access Policy 39 (IMR15_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR15 Write Access Policy 38 (IMR15_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR15 Write Access Policy 37 (IMR15_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR15 Write Access Policy 36 (IMR15_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR15 Write Access Policy 35 (IMR15_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR15 Write Access Policy 34 (IMR15_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR15 Write Access Policy 33 (IMR15_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR15 Write Access Policy 32 (IMR15_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR15 Write Access Policy 31 (IMR15_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR15 Write Access Policy 30 (IMR15_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR15 Write Access Policy 29 (IMR15_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR15 Write Access Policy 28 (IMR15_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR15 Write Access Policy 27 (IMR15_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR15 Write Access Policy 26 (IMR15_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR15 Write Access Policy 25 (IMR15_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR15 Write Access Policy 24 (IMR15_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR15 Write Access Policy 23 (IMR15_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR15 Write Access Policy 22 (IMR15_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR15 Write Access Policy 21 (IMR15_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**IMR15 Write Access Policy 20 (IMR15_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR15 Write Access Policy 19 (IMR15_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR15 Write Access Policy 18 (IMR15_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR15 Write Access Policy 17 (IMR15_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR15 Write Access Policy 16 (IMR15_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR15 Write Access Policy 15 (IMR15_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR15 Write Access Policy 14 (IMR15_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR15 Write Access Policy 13 (IMR15_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR15 Write Access Policy 12 (IMR15_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR15 Write Access Policy 11 (IMR15_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR15 Write Access Policy 10 (IMR15_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR15 Write Access Policy 9 (IMR15_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR15 Write Access Policy 8 (IMR15_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR15 Write Access Policy 7 (IMR15_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR15 Write Access Policy 6 (IMR15_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR15 Write Access Policy 5 (IMR15_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR15 Write Access Policy 4 (IMR15_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR15 Write Access Policy 3 (IMR15_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR15 Write Access Policy 2 (IMR15_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**IMR15 Write Access Policy 1 (IMR15_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR15 Write Access Policy 0 (IMR15_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR15 region, based on each agent's 6bit encoded SAI value.

3.368 IMR16 Base (B_CR_BIMR16BASE_0_0_0_MCHBAR) — Offset 6A70h

This register, along with IMR16MASK, IMR16RAC, and IMR16WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR16RAC and IMR16WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6A70h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR16 Base (IMR16_BASE):** Specifies bits 38:10 of the start address of IMR16 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR16MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR16 defined region.

3.369 IMR16 Mask (B_CR_BIMR16MASK_0_0_0_MCHBAR) — Offset 6A74h

This register, along with IMR16BASE, IMR16RAC, and IMR16WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR16RAC and IMR16WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6A74h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved
28:0	0h RW	**Mask 0 IMR16 Mask (IMR16_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR16BASE[28:0] value. A match indicates that the incoming address falls within the IMR16 region.

3.370 IMR16 Control Policy (B_CR_BIMR16CP_0_0_0_MCHBAR) — Offset 6A78h

This register controls the access policy to the Read Access Policy BIMR16RAC, Write Access Policy BIMR16WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6A78h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
60	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
59	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
58	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
55	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
52	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
50	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
47	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
44	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR16 Control Policy (IMR16_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR16 Control Policy (IMR16_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
41	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
40	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
39	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
36	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
33	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
31	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR16 Control Policy (IMR16_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR16 Control Policy (IMR16_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
28	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
25	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR16 Control Policy (IMR16_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
22	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
21	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
17	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR16 Control Policy (IMR16_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
14	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
12	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
9	1h RW	**IMR16 Control Policy (IMR16_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
6	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
3	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
2	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
1	1h RW	**IMR16 Control Policy (IMR16_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR16 Control Policy (IMR16_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR16RAC, BIMR16WAC and BIMR16CP registers based on the value from each agent's 6bit SAI field.

3.371 IMR16 Read Access Policy (B_CR_BIMR16RAC_0_0_0_MCHBAR) — Offset 6A80h

This register, along with IMR16BASE, IMR16MASK and IMR16WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR16. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6A80h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR16 Read Access Policy 63 (IMR16_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR16 Read Access Policy 62 (IMR16_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR16 Read Access Policy 61 (IMR16_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR16 Read Access Policy 60 (IMR16_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR16 Read Access Policy 59 (IMR16_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR16 Read Access Policy 58 (IMR16_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR16 Read Access Policy 57 (IMR16_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR16 Read Access Policy 56 (IMR16_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR16 Read Access Policy 55 (IMR16_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR16 Read Access Policy 54 (IMR16_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
53	0h RO	**IMR16 Read Access Policy 53 (IMR16_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR16 Read Access Policy 52 (IMR16_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR16 Read Access Policy 51 (IMR16_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR16 Read Access Policy 50 (IMR16_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR16 Read Access Policy 49 (IMR16_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR16 Read Access Policy 48 (IMR16_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR16 Read Access Policy 47 (IMR16_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR16 Read Access Policy 46 (IMR16_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR16 Read Access Policy 45 (IMR16_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR16 Read Access Policy 44 (IMR16_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR16 Read Access Policy 43 (IMR16_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR16 Read Access Policy 42 (IMR16_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR16 Read Access Policy 41 (IMR16_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR16 Read Access Policy 40 (IMR16_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR16 Read Access Policy 39 (IMR16_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR16 Read Access Policy 38 (IMR16_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR16 Read Access Policy 37 (IMR16_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR16 Read Access Policy 36 (IMR16_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR16 Read Access Policy 35 (IMR16_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
34	0h RW	**IMR16 Read Access Policy 34 (IMR16_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR16 Read Access Policy 33 (IMR16_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR16 Read Access Policy 32 (IMR16_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR16 Read Access Policy 31 (IMR16_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR16 Read Access Policy 30 (IMR16_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR16 Read Access Policy 29 (IMR16_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR16 Read Access Policy 28 (IMR16_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR16 Read Access Policy 27 (IMR16_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR16 Read Access Policy 26 (IMR16_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR16 Read Access Policy 25 (IMR16_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR16 Read Access Policy 24 (IMR16_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR16 Read Access Policy 23 (IMR16_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR16 Read Access Policy 22 (IMR16_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR16 Read Access Policy 21 (IMR16_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR16 Read Access Policy 20 (IMR16_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR16 Read Access Policy 19 (IMR16_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR16 Read Access Policy 18 (IMR16_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR16 Read Access Policy 17 (IMR16_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR16 Read Access Policy 16 (IMR16_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**IMR16 Read Access Policy 15 (IMR16_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR16 Read Access Policy 14 (IMR16_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR16 Read Access Policy 13 (IMR16_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR16 Read Access Policy 12 (IMR16_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR16 Read Access Policy 11 (IMR16_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR16 Read Access Policy 10 (IMR16_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR16 Read Access Policy 9 (IMR16_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR16 Read Access Policy 8 (IMR16_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR16 Read Access Policy 7 (IMR16_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR16 Read Access Policy 6 (IMR16_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR16 Read Access Policy 5 (IMR16_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR16 Read Access Policy 4 (IMR16_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR16 Read Access Policy 3 (IMR16_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR16 Read Access Policy 2 (IMR16_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR16 Read Access Policy 1 (IMR16_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR16 Read Access Policy 0 (IMR16_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR16 region, based on each agent's 6bit encoded SAI value.

3.372 IMR16 Write Access Policy (B_CR_BIMR16WAC_0_0_0_MCHBAR) — Offset 6A88h

This register, along with IMR16BASE, IMR16MASK and IMR16RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR16. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6A88h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR16 Write Access Policy 63 (IMR16_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR16 Write Access Policy 62 (IMR16_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR16 Write Access Policy 61 (IMR16_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR16 Write Access Policy 60 (IMR16_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR16 Write Access Policy 59 (IMR16_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR16 Write Access Policy 58 (IMR16_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR16 Write Access Policy 57 (IMR16_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR16 Write Access Policy 56 (IMR16_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR16 Write Access Policy 55 (IMR16_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR16 Write Access Policy 54 (IMR16_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR16 Write Access Policy 53 (IMR16_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR16 Write Access Policy 52 (IMR16_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR16 Write Access Policy 51 (IMR16_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
50	0h RW	**IMR16 Write Access Policy 50 (IMR16_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR16 Write Access Policy 49 (IMR16_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR16 Write Access Policy 48 (IMR16_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR16 Write Access Policy 47 (IMR16_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR16 Write Access Policy 46 (IMR16_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR16 Write Access Policy 45 (IMR16_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR16 Write Access Policy 44 (IMR16_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR16 Write Access Policy 43 (IMR16_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR16 Write Access Policy 42 (IMR16_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR16 Write Access Policy 41 (IMR16_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR16 Write Access Policy 40 (IMR16_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR16 Write Access Policy 39 (IMR16_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR16 Write Access Policy 38 (IMR16_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR16 Write Access Policy 37 (IMR16_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR16 Write Access Policy 36 (IMR16_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR16 Write Access Policy 35 (IMR16_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR16 Write Access Policy 34 (IMR16_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR16 Write Access Policy 33 (IMR16_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR16 Write Access Policy 32 (IMR16_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**IMR16 Write Access Policy 31 (IMR16_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR16 Write Access Policy 30 (IMR16_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR16 Write Access Policy 29 (IMR16_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR16 Write Access Policy 28 (IMR16_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR16 Write Access Policy 27 (IMR16_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR16 Write Access Policy 26 (IMR16_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR16 Write Access Policy 25 (IMR16_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR16 Write Access Policy 24 (IMR16_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR16 Write Access Policy 23 (IMR16_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR16 Write Access Policy 22 (IMR16_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR16 Write Access Policy 21 (IMR16_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR16 Write Access Policy 20 (IMR16_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR16 Write Access Policy 19 (IMR16_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR16 Write Access Policy 18 (IMR16_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR16 Write Access Policy 17 (IMR16_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR16 Write Access Policy 16 (IMR16_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR16 Write Access Policy 15 (IMR16_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR16 Write Access Policy 14 (IMR16_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR16 Write Access Policy 13 (IMR16_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**IMR16 Write Access Policy 12 (IMR16_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR16 Write Access Policy 11 (IMR16_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR16 Write Access Policy 10 (IMR16_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR16 Write Access Policy 9 (IMR16_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR16 Write Access Policy 8 (IMR16_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR16 Write Access Policy 7 (IMR16_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR16 Write Access Policy 6 (IMR16_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR16 Write Access Policy 5 (IMR16_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR16 Write Access Policy 4 (IMR16_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR16 Write Access Policy 3 (IMR16_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR16 Write Access Policy 2 (IMR16_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR16 Write Access Policy 1 (IMR16_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR16 Write Access Policy 0 (IMR16_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR16 region, based on each agent's 6bit encoded SAI value.

3.373 IMR17 Base (B_CR_BIMR17BASE_0_0_0_MCHBAR) — Offset 6A90h

This register, along with IMR17MASK, IMR17RAC, and IMR17WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR17RAC and IMR17WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6A90h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR17 Base (IMR17_BASE):** Specifies bits 38:10 of the start address of IMR17 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR17MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR17 defined region.

3.374 IMR17 Mask (B_CR_BIMR17MASK_0_0_0_MCHBAR) — Offset 6A94h

This register, along with IMR17BASE, IMR17RAC, and IMR17WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR17RAC and IMR17WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6A94h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved
28:0	0h RW	**Mask 0 IMR17 Mask (IMR17_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR17BASE[28:0] value. A match indicates that the incoming address falls within the IMR17 region.

3.375 IMR17 Control Policy (B_CR_BIMR17CP_0_0_0_MCHBAR) — Offset 6A98h

This register controls the access policy to the Read Access Policy BIMR17RAC, Write Access Policy BIMR17WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6A98h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
60	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
59	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
58	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
55	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
52	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
50	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
47	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
44	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR17 Control Policy (IMR17_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR17 Control Policy (IMR17_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
41	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
40	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
39	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
36	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
33	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR17 Control Policy (IMR17_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR17 Control Policy (IMR17_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
28	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
25	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR17 Control Policy (IMR17_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
22	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
21	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
20	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
17	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR17 Control Policy (IMR17_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
14	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
9	1h RW	**IMR17 Control Policy (IMR17_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
6	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
3	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
2	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
1	1h RW	**IMR17 Control Policy (IMR17_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR17 Control Policy (IMR17_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR17RAC, BIMR17WAC and BIMR17CP registers based on the value from each agent's 6bit SAI field.

3.376 IMR17 Read Access Policy (B_CR_BIMR17RAC_0_0_0_MCHBAR) — Offset 6AA0h

This register, along with IMR17BASE, IMR17MASK and IMR17WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR17. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6AA0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR17 Read Access Policy 63 (IMR17_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR17 Read Access Policy 62 (IMR17_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR17 Read Access Policy 61 (IMR17_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR17 Read Access Policy 60 (IMR17_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR17 Read Access Policy 59 (IMR17_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR17 Read Access Policy 58 (IMR17_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR17 Read Access Policy 57 (IMR17_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR17 Read Access Policy 56 (IMR17_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR17 Read Access Policy 55 (IMR17_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR17 Read Access Policy 54 (IMR17_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR17 Read Access Policy 53 (IMR17_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR17 Read Access Policy 52 (IMR17_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR17 Read Access Policy 51 (IMR17_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR17 Read Access Policy 50 (IMR17_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR17 Read Access Policy 49 (IMR17_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR17 Read Access Policy 48 (IMR17_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR17 Read Access Policy 47 (IMR17_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR17 Read Access Policy 46 (IMR17_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR17 Read Access Policy 45 (IMR17_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
44	0h RW	**IMR17 Read Access Policy 44 (IMR17_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR17 Read Access Policy 43 (IMR17_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR17 Read Access Policy 42 (IMR17_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR17 Read Access Policy 41 (IMR17_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR17 Read Access Policy 40 (IMR17_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR17 Read Access Policy 39 (IMR17_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR17 Read Access Policy 38 (IMR17_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR17 Read Access Policy 37 (IMR17_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR17 Read Access Policy 36 (IMR17_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR17 Read Access Policy 35 (IMR17_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR17 Read Access Policy 34 (IMR17_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR17 Read Access Policy 33 (IMR17_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR17 Read Access Policy 32 (IMR17_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR17 Read Access Policy 31 (IMR17_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR17 Read Access Policy 30 (IMR17_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR17 Read Access Policy 29 (IMR17_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR17 Read Access Policy 28 (IMR17_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR17 Read Access Policy 27 (IMR17_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR17 Read Access Policy 26 (IMR17_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**IMR17 Read Access Policy 25 (IMR17_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR17 Read Access Policy 24 (IMR17_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR17 Read Access Policy 23 (IMR17_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR17 Read Access Policy 22 (IMR17_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR17 Read Access Policy 21 (IMR17_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR17 Read Access Policy 20 (IMR17_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR17 Read Access Policy 19 (IMR17_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR17 Read Access Policy 18 (IMR17_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR17 Read Access Policy 17 (IMR17_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR17 Read Access Policy 16 (IMR17_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR17 Read Access Policy 15 (IMR17_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR17 Read Access Policy 14 (IMR17_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR17 Read Access Policy 13 (IMR17_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR17 Read Access Policy 12 (IMR17_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR17 Read Access Policy 11 (IMR17_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR17 Read Access Policy 10 (IMR17_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR17 Read Access Policy 9 (IMR17_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR17 Read Access Policy 8 (IMR17_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR17 Read Access Policy 7 (IMR17_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**IMR17 Read Access Policy 6 (IMR17_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR17 Read Access Policy 5 (IMR17_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR17 Read Access Policy 4 (IMR17_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR17 Read Access Policy 3 (IMR17_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR17 Read Access Policy 2 (IMR17_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR17 Read Access Policy 1 (IMR17_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR17 Read Access Policy 0 (IMR17_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR17 region, based on each agent's 6bit encoded SAI value.

3.377 IMR17 Write Access Policy (B_CR_BIMR17WAC_0_0_0_MCHBAR) — Offset 6AA8h

This register, along with IMR17BASE, IMR17MASK and IMR17RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR17. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6AA8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR17 Write Access Policy 63 (IMR17_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR17 Write Access Policy 62 (IMR17_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR17 Write Access Policy 61 (IMR17_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR17 Write Access Policy 60 (IMR17_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR17 Write Access Policy 59 (IMR17_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
58	0h RO	**IMR17 Write Access Policy 58 (IMR17_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR17 Write Access Policy 57 (IMR17_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR17 Write Access Policy 56 (IMR17_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR17 Write Access Policy 55 (IMR17_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR17 Write Access Policy 54 (IMR17_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR17 Write Access Policy 53 (IMR17_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR17 Write Access Policy 52 (IMR17_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR17 Write Access Policy 51 (IMR17_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR17 Write Access Policy 50 (IMR17_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR17 Write Access Policy 49 (IMR17_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR17 Write Access Policy 48 (IMR17_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR17 Write Access Policy 47 (IMR17_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR17 Write Access Policy 46 (IMR17_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR17 Write Access Policy 45 (IMR17_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR17 Write Access Policy 44 (IMR17_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR17 Write Access Policy 43 (IMR17_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR17 Write Access Policy 42 (IMR17_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR17 Write Access Policy 41 (IMR17_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR17 Write Access Policy 40 (IMR17_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
39	0h RO	**IMR17 Write Access Policy 39 (IMR17_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR17 Write Access Policy 38 (IMR17_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR17 Write Access Policy 37 (IMR17_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR17 Write Access Policy 36 (IMR17_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR17 Write Access Policy 35 (IMR17_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR17 Write Access Policy 34 (IMR17_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR17 Write Access Policy 33 (IMR17_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR17 Write Access Policy 32 (IMR17_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR17 Write Access Policy 31 (IMR17_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR17 Write Access Policy 30 (IMR17_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR17 Write Access Policy 29 (IMR17_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR17 Write Access Policy 28 (IMR17_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR17 Write Access Policy 27 (IMR17_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR17 Write Access Policy 26 (IMR17_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR17 Write Access Policy 25 (IMR17_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR17 Write Access Policy 24 (IMR17_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR17 Write Access Policy 23 (IMR17_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR17 Write Access Policy 22 (IMR17_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR17 Write Access Policy 21 (IMR17_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**IMR17 Write Access Policy 20 (IMR17_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR17 Write Access Policy 19 (IMR17_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR17 Write Access Policy 18 (IMR17_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR17 Write Access Policy 17 (IMR17_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR17 Write Access Policy 16 (IMR17_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR17 Write Access Policy 15 (IMR17_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR17 Write Access Policy 14 (IMR17_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR17 Write Access Policy 13 (IMR17_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR17 Write Access Policy 12 (IMR17_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR17 Write Access Policy 11 (IMR17_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR17 Write Access Policy 10 (IMR17_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR17 Write Access Policy 9 (IMR17_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR17 Write Access Policy 8 (IMR17_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR17 Write Access Policy 7 (IMR17_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR17 Write Access Policy 6 (IMR17_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR17 Write Access Policy 5 (IMR17_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR17 Write Access Policy 4 (IMR17_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR17 Write Access Policy 3 (IMR17_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR17 Write Access Policy 2 (IMR17_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**IMR17 Write Access Policy 1 (IMR17_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR17 Write Access Policy 0 (IMR17_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR17 region, based on each agent's 6bit encoded SAI value.

3.378 IMR18 Base (B_CR_BIMR18BASE_0_0_0_MCHBAR) — Offset 6AB0h

This register, along with IMR18MASK, IMR18RAC, and IMR18WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR18RAC and IMR18WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6AB0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR18 Base (IMR18_BASE):** Specifies bits 38:10 of the start address of IMR18 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR18MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR18 defined region.

3.379 IMR18 Mask (B_CR_BIMR18MASK_0_0_0_MCHBAR) — Offset 6AB4h

This register, along with IMR18BASE, IMR18RAC, and IMR18WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR18RAC and IMR18WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6AB4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved
28:0	0h RW	**Mask 0 IMR18 Mask (IMR18_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR18BASE[28:0] value. A match indicates that the incoming address falls within the IMR18 region.

3.380 IMR18 Control Policy (B_CR_BIMR18CP_0_0_0_MCHBAR) — Offset 6AB8h

This register controls the access policy to the Read Access Policy BIMR18RAC, Write Access Policy BIMR18WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6AB8h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
60	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
59	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
58	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
55	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
52	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
50	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
47	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
44	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR18 Control Policy (IMR18_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR18 Control Policy (IMR18_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
41	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
40	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
39	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
36	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
33	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
31	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR18 Control Policy (IMR18_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR18 Control Policy (IMR18_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
28	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
25	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR18 Control Policy (IMR18_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
22	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
21	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
17	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR18 Control Policy (IMR18_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
14	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
12	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
9	1h RW	**IMR18 Control Policy (IMR18_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
6	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
3	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
2	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
1	1h RW	**IMR18 Control Policy (IMR18_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR18 Control Policy (IMR18_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR18RAC, BIMR18WAC and BIMR18CP registers based on the value from each agent's 6bit SAI field.

3.381 IMR18 Read Access Policy (B_CR_BIMR18RAC_0_0_0_MCHBAR) — Offset 6AC0h

This register, along with IMR18BASE, IMR18MASK and IMR18WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR18. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6AC0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR18 Read Access Policy 63 (IMR18_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR18 Read Access Policy 62 (IMR18_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR18 Read Access Policy 61 (IMR18_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR18 Read Access Policy 60 (IMR18_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR18 Read Access Policy 59 (IMR18_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR18 Read Access Policy 58 (IMR18_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR18 Read Access Policy 57 (IMR18_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR18 Read Access Policy 56 (IMR18_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR18 Read Access Policy 55 (IMR18_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR18 Read Access Policy 54 (IMR18_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
53	0h RO	**IMR18 Read Access Policy 53 (IMR18_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR18 Read Access Policy 52 (IMR18_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR18 Read Access Policy 51 (IMR18_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR18 Read Access Policy 50 (IMR18_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR18 Read Access Policy 49 (IMR18_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR18 Read Access Policy 48 (IMR18_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR18 Read Access Policy 47 (IMR18_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR18 Read Access Policy 46 (IMR18_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR18 Read Access Policy 45 (IMR18_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR18 Read Access Policy 44 (IMR18_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR18 Read Access Policy 43 (IMR18_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR18 Read Access Policy 42 (IMR18_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR18 Read Access Policy 41 (IMR18_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR18 Read Access Policy 40 (IMR18_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR18 Read Access Policy 39 (IMR18_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR18 Read Access Policy 38 (IMR18_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR18 Read Access Policy 37 (IMR18_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR18 Read Access Policy 36 (IMR18_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR18 Read Access Policy 35 (IMR18_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
34	0h RW	**IMR18 Read Access Policy 34 (IMR18_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR18 Read Access Policy 33 (IMR18_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR18 Read Access Policy 32 (IMR18_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR18 Read Access Policy 31 (IMR18_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR18 Read Access Policy 30 (IMR18_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR18 Read Access Policy 29 (IMR18_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR18 Read Access Policy 28 (IMR18_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR18 Read Access Policy 27 (IMR18_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR18 Read Access Policy 26 (IMR18_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR18 Read Access Policy 25 (IMR18_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR18 Read Access Policy 24 (IMR18_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR18 Read Access Policy 23 (IMR18_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR18 Read Access Policy 22 (IMR18_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR18 Read Access Policy 21 (IMR18_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR18 Read Access Policy 20 (IMR18_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR18 Read Access Policy 19 (IMR18_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR18 Read Access Policy 18 (IMR18_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR18 Read Access Policy 17 (IMR18_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR18 Read Access Policy 16 (IMR18_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**IMR18 Read Access Policy 15 (IMR18_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR18 Read Access Policy 14 (IMR18_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR18 Read Access Policy 13 (IMR18_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR18 Read Access Policy 12 (IMR18_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR18 Read Access Policy 11 (IMR18_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR18 Read Access Policy 10 (IMR18_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR18 Read Access Policy 9 (IMR18_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR18 Read Access Policy 8 (IMR18_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR18 Read Access Policy 7 (IMR18_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR18 Read Access Policy 6 (IMR18_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR18 Read Access Policy 5 (IMR18_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR18 Read Access Policy 4 (IMR18_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR18 Read Access Policy 3 (IMR18_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR18 Read Access Policy 2 (IMR18_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR18 Read Access Policy 1 (IMR18_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR18 Read Access Policy 0 (IMR18_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR18 region, based on each agent's 6bit encoded SAI value.

3.382 IMR18 Write Access Policy (B_CR_BIMR18WAC_0_0_0_MCHBAR) — Offset 6AC8h

This register, along with IMR18BASE, IMR18MASK and IMR18RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR18. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6AC8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR18 Write Access Policy 63 (IMR18_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR18 Write Access Policy 62 (IMR18_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR18 Write Access Policy 61 (IMR18_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR18 Write Access Policy 60 (IMR18_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR18 Write Access Policy 59 (IMR18_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR18 Write Access Policy 58 (IMR18_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR18 Write Access Policy 57 (IMR18_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR18 Write Access Policy 56 (IMR18_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR18 Write Access Policy 55 (IMR18_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR18 Write Access Policy 54 (IMR18_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR18 Write Access Policy 53 (IMR18_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR18 Write Access Policy 52 (IMR18_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR18 Write Access Policy 51 (IMR18_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
50	0h RW	**IMR18 Write Access Policy 50 (IMR18_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR18 Write Access Policy 49 (IMR18_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR18 Write Access Policy 48 (IMR18_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR18 Write Access Policy 47 (IMR18_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR18 Write Access Policy 46 (IMR18_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR18 Write Access Policy 45 (IMR18_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR18 Write Access Policy 44 (IMR18_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR18 Write Access Policy 43 (IMR18_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR18 Write Access Policy 42 (IMR18_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR18 Write Access Policy 41 (IMR18_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR18 Write Access Policy 40 (IMR18_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR18 Write Access Policy 39 (IMR18_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR18 Write Access Policy 38 (IMR18_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR18 Write Access Policy 37 (IMR18_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR18 Write Access Policy 36 (IMR18_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR18 Write Access Policy 35 (IMR18_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR18 Write Access Policy 34 (IMR18_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR18 Write Access Policy 33 (IMR18_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR18 Write Access Policy 32 (IMR18_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**IMR18 Write Access Policy 31 (IMR18_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR18 Write Access Policy 30 (IMR18_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR18 Write Access Policy 29 (IMR18_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR18 Write Access Policy 28 (IMR18_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR18 Write Access Policy 27 (IMR18_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR18 Write Access Policy 26 (IMR18_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR18 Write Access Policy 25 (IMR18_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR18 Write Access Policy 24 (IMR18_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR18 Write Access Policy 23 (IMR18_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR18 Write Access Policy 22 (IMR18_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR18 Write Access Policy 21 (IMR18_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR18 Write Access Policy 20 (IMR18_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR18 Write Access Policy 19 (IMR18_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR18 Write Access Policy 18 (IMR18_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR18 Write Access Policy 17 (IMR18_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR18 Write Access Policy 16 (IMR18_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR18 Write Access Policy 15 (IMR18_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR18 Write Access Policy 14 (IMR18_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR18 Write Access Policy 13 (IMR18_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**IMR18 Write Access Policy 12 (IMR18_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR18 Write Access Policy 11 (IMR18_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR18 Write Access Policy 10 (IMR18_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR18 Write Access Policy 9 (IMR18_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR18 Write Access Policy 8 (IMR18_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR18 Write Access Policy 7 (IMR18_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR18 Write Access Policy 6 (IMR18_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR18 Write Access Policy 5 (IMR18_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR18 Write Access Policy 4 (IMR18_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR18 Write Access Policy 3 (IMR18_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR18 Write Access Policy 2 (IMR18_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR18 Write Access Policy 1 (IMR18_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR18 Write Access Policy 0 (IMR18_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR18 region, based on each agent's 6bit encoded SAI value.

3.383 IMR19 Base (B_CR_BIMR19BASE_0_0_0_MCHBAR) — Offset 6AD0h

This register, along with IMR19MASK, IMR19RAC, and IMR19WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR19RAC and IMR19WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6AD0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the IMR region.
30	0h RW	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:0	0h RW	**Base 0 IMR19 Base (IMR19_BASE):** Specifies bits 38:10 of the start address of IMR19 region. IMR region size must be a strict poweroftwo, at least 1KB, and naturally aligned to the size. These bits are compared with the result of the IMR19MASK[28:0] applied to bits 38:10 of the incoming address, to determine if an access falls within the IMR19 defined region.

3.384 IMR19 Mask (B_CR_BIMR19MASK_0_0_0_MCHBAR) — Offset 6AD4h

This register, along with IMR19BASE, IMR19RAC, and IMR19WAC, defines an isolated region of memory that can be masked to prohibit certain system agents from accessing memory. When an agent sends a request to the B-Unit, whether snooped or not, an IMR may optionally prevent that transaction from changing the state of memory or from getting correct data in response to the operation, if the agent's SAI field does not specify the correct Policy. The IMR's Policy is configured by the IMR19RAC and IMR19WAC registers.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6AD4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables return to the requester of implicit writeback HITM data from GT. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_0):** Reserved
28:0	0h RW	**Mask 0 IMR19 Mask (IMR19_MASK):** These bits are ANDed with bits 38:10 of the incoming address to determine if the combined result matches the IMR19BASE[28:0] value. A match indicates that the incoming address falls within the IMR19 region.

3.385 IMR19 Control Policy (B_CR_BIMR19CP_0_0_0_MCHBAR) — Offset 6AD8h

This register controls the access policy to the Read Access Policy BIMR19RAC, Write Access Policy BIMR19WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6AD8h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
62	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
61	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
60	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
59	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
58	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
57	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
56	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
55	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
54	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
53	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
52	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
51	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
50	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
49	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
48	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
47	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
46	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
45	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
44	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
43	1h RW	**IMR19 Control Policy (IMR19_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
42	1h RW	**IMR19 Control Policy (IMR19_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
41	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
40	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
39	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
38	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
37	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
36	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
35	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
34	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
33	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
32	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
30	1h RW	**IMR19 Control Policy (IMR19_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
29	1h RW	**IMR19 Control Policy (IMR19_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
28	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
27	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
26	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
25	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
24	1h RW	**IMR19 Control Policy (IMR19_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
23	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
22	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
21	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
20	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
19	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
18	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
17	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
16	1h RW	**IMR19 Control Policy (IMR19_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
15	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
14	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
13	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
11	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
10	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
9	1h RW	**IMR19 Control Policy (IMR19_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
8	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
7	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
6	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
5	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
4	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
3	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
2	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
1	1h RW	**IMR19 Control Policy (IMR19_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.
0	0h RW	**IMR19 Control Policy (IMR19_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BIMR19RAC, BIMR19WAC and BIMR19CP registers based on the value from each agent's 6bit SAI field.

3.386 IMR19 Read Access Policy (B_CR_BIMR19RAC_0_0_0_MCHBAR) — Offset 6AE0h

This register, along with IMR19BASE, IMR19MASK and IMR19WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR19. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6AE0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR19 Read Access Policy 63 (IMR19_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR19 Read Access Policy 62 (IMR19_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR19 Read Access Policy 61 (IMR19_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR19 Read Access Policy 60 (IMR19_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR19 Read Access Policy 59 (IMR19_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
58	0h RO	**IMR19 Read Access Policy 58 (IMR19_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR19 Read Access Policy 57 (IMR19_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR19 Read Access Policy 56 (IMR19_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR19 Read Access Policy 55 (IMR19_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR19 Read Access Policy 54 (IMR19_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR19 Read Access Policy 53 (IMR19_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR19 Read Access Policy 52 (IMR19_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR19 Read Access Policy 51 (IMR19_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR19 Read Access Policy 50 (IMR19_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR19 Read Access Policy 49 (IMR19_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR19 Read Access Policy 48 (IMR19_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR19 Read Access Policy 47 (IMR19_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR19 Read Access Policy 46 (IMR19_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR19 Read Access Policy 45 (IMR19_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
44	0h RW	**IMR19 Read Access Policy 44 (IMR19_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR19 Read Access Policy 43 (IMR19_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR19 Read Access Policy 42 (IMR19_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR19 Read Access Policy 41 (IMR19_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR19 Read Access Policy 40 (IMR19_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
39	0h RO	**IMR19 Read Access Policy 39 (IMR19_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR19 Read Access Policy 38 (IMR19_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR19 Read Access Policy 37 (IMR19_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR19 Read Access Policy 36 (IMR19_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR19 Read Access Policy 35 (IMR19_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR19 Read Access Policy 34 (IMR19_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR19 Read Access Policy 33 (IMR19_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR19 Read Access Policy 32 (IMR19_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR19 Read Access Policy 31 (IMR19_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR19 Read Access Policy 30 (IMR19_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR19 Read Access Policy 29 (IMR19_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR19 Read Access Policy 28 (IMR19_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR19 Read Access Policy 27 (IMR19_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR19 Read Access Policy 26 (IMR19_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**IMR19 Read Access Policy 25 (IMR19_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR19 Read Access Policy 24 (IMR19_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR19 Read Access Policy 23 (IMR19_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR19 Read Access Policy 22 (IMR19_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR19 Read Access Policy 21 (IMR19_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
20	0h RO	**IMR19 Read Access Policy 20 (IMR19_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR19 Read Access Policy 19 (IMR19_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR19 Read Access Policy 18 (IMR19_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR19 Read Access Policy 17 (IMR19_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR19 Read Access Policy 16 (IMR19_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR19 Read Access Policy 15 (IMR19_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR19 Read Access Policy 14 (IMR19_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR19 Read Access Policy 13 (IMR19_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR19 Read Access Policy 12 (IMR19_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR19 Read Access Policy 11 (IMR19_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR19 Read Access Policy 10 (IMR19_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR19 Read Access Policy 9 (IMR19_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR19 Read Access Policy 8 (IMR19_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR19 Read Access Policy 7 (IMR19_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**IMR19 Read Access Policy 6 (IMR19_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR19 Read Access Policy 5 (IMR19_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR19 Read Access Policy 4 (IMR19_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR19 Read Access Policy 3 (IMR19_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR19 Read Access Policy 2 (IMR19_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
1	0h RW	**IMR19 Read Access Policy 1 (IMR19_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR19 Read Access Policy 0 (IMR19_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the IMR19 region, based on each agent's 6bit encoded SAI value.

3.387 IMR19 Write Access Policy (B_CR_BIMR19WAC_0_0_0_MCHBAR) — Offset 6AE8h

This register, along with IMR19BASE, IMR19MASK and IMR19RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to IMR19. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6AE8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**IMR19 Write Access Policy 63 (IMR19_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
62	0h RO	**IMR19 Write Access Policy 62 (IMR19_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
61	0h RO	**IMR19 Write Access Policy 61 (IMR19_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
60	0h RO	**IMR19 Write Access Policy 60 (IMR19_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
59	0h RW	**IMR19 Write Access Policy 59 (IMR19_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
58	0h RO	**IMR19 Write Access Policy 58 (IMR19_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
57	0h RO	**IMR19 Write Access Policy 57 (IMR19_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
56	0h RW	**IMR19 Write Access Policy 56 (IMR19_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
55	0h RW	**IMR19 Write Access Policy 55 (IMR19_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
54	0h RW	**IMR19 Write Access Policy 54 (IMR19_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
53	0h RO	**IMR19 Write Access Policy 53 (IMR19_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
52	0h RO	**IMR19 Write Access Policy 52 (IMR19_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
51	0h RO	**IMR19 Write Access Policy 51 (IMR19_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
50	0h RW	**IMR19 Write Access Policy 50 (IMR19_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
49	0h RW	**IMR19 Write Access Policy 49 (IMR19_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
48	0h RW	**IMR19 Write Access Policy 48 (IMR19_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
47	0h RO	**IMR19 Write Access Policy 47 (IMR19_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
46	0h RO	**IMR19 Write Access Policy 46 (IMR19_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
45	0h RO	**IMR19 Write Access Policy 45 (IMR19_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
44	0h RW	**IMR19 Write Access Policy 44 (IMR19_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
43	0h RW	**IMR19 Write Access Policy 43 (IMR19_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
42	0h RW	**IMR19 Write Access Policy 42 (IMR19_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
41	0h RW	**IMR19 Write Access Policy 41 (IMR19_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
40	0h RW	**IMR19 Write Access Policy 40 (IMR19_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
39	0h RO	**IMR19 Write Access Policy 39 (IMR19_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
38	0h RW	**IMR19 Write Access Policy 38 (IMR19_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
37	0h RO	**IMR19 Write Access Policy 37 (IMR19_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
36	0h RW	**IMR19 Write Access Policy 36 (IMR19_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
35	0h RO	**IMR19 Write Access Policy 35 (IMR19_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
34	0h RW	**IMR19 Write Access Policy 34 (IMR19_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
33	0h RW	**IMR19 Write Access Policy 33 (IMR19_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
32	0h RW	**IMR19 Write Access Policy 32 (IMR19_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
31	0h RO	**IMR19 Write Access Policy 31 (IMR19_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
30	0h RW	**IMR19 Write Access Policy 30 (IMR19_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
29	0h RW	**IMR19 Write Access Policy 29 (IMR19_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
28	0h RW	**IMR19 Write Access Policy 28 (IMR19_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
27	0h RW	**IMR19 Write Access Policy 27 (IMR19_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
26	0h RW	**IMR19 Write Access Policy 26 (IMR19_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
25	0h RW	**IMR19 Write Access Policy 25 (IMR19_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
24	0h RW	**IMR19 Write Access Policy 24 (IMR19_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
23	0h RO	**IMR19 Write Access Policy 23 (IMR19_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
22	0h RO	**IMR19 Write Access Policy 22 (IMR19_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
21	0h RW	**IMR19 Write Access Policy 21 (IMR19_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**IMR19 Write Access Policy 20 (IMR19_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
19	0h RO	**IMR19 Write Access Policy 19 (IMR19_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
18	0h RO	**IMR19 Write Access Policy 18 (IMR19_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
17	0h RW	**IMR19 Write Access Policy 17 (IMR19_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
16	0h RW	**IMR19 Write Access Policy 16 (IMR19_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
15	0h RO	**IMR19 Write Access Policy 15 (IMR19_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
14	0h RO	**IMR19 Write Access Policy 14 (IMR19_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
13	0h RW	**IMR19 Write Access Policy 13 (IMR19_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
12	0h RW	**IMR19 Write Access Policy 12 (IMR19_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
11	0h RO	**IMR19 Write Access Policy 11 (IMR19_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
10	0h RO	**IMR19 Write Access Policy 10 (IMR19_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
9	0h RO	**IMR19 Write Access Policy 9 (IMR19_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
8	0h RW	**IMR19 Write Access Policy 8 (IMR19_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
7	0h RW	**IMR19 Write Access Policy 7 (IMR19_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
6	0h RW	**IMR19 Write Access Policy 6 (IMR19_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
5	0h RW	**IMR19 Write Access Policy 5 (IMR19_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
4	0h RW	**IMR19 Write Access Policy 4 (IMR19_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
3	0h RW	**IMR19 Write Access Policy 3 (IMR19_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
2	0h RW	**IMR19 Write Access Policy 2 (IMR19_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**IMR19 Write Access Policy 1 (IMR19_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.
0	0h RO	**IMR19 Write Access Policy 0 (IMR19_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the IMR19 region, based on each agent's 6bit encoded SAI value.

3.388 MOT Out Base (B_CR_MOT_OUT_BASE_0_0_0_MCHBAR) — Offset 6AF0h

This register contains the value of the start address of the MOT debug data region. The smallest reserved region for MOT debug data, if enabled, is 16MB. The MOT region must be power of two sized and naturally aligned to its size. The MOT region is evenly distributed between the slices and interleaved on a 4K granularity. When generating addresses, B-Unit MOT H/W within each slice ensures that the addresses it generates are within the region mapped to that slice.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6AF0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**IMR Enable (IMR_EN):** Enables access checking for the MOT region. Note: this does not enable MOT itself, but merely enables access control checks for transactions that attempt to access the MOT buffer.
30	0h RO	**Asset Classification AC[0]: Trace Enable (TR_EN):** Enables snooping of transactions to the IMR region by tracing agents such as MOT. Reserved and set to 0 for the MOT region, since otherwise this would enable recursive tracing and corrupt MOT buffer.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:14	0h RW	**MOT_OUT Base (MOT_OUT_BASE):** Specifies bits 38:24 of the start address of the MOT memory region. Region size must be a strict power of two, at least 16MB, and naturally aligned to the size. These bits are compared with the result of the MOT_OUT_MASK[28:14] applied to bits 38:24 of the incoming address to determine if an access falls within the MOT region.
13:0	0h RO	**Reserved (RESERVED_0):** Reserved

3.389 MOT Out Mask (B_CR_MOT_OUT_MASK_0_0_0_MCHBAR) — Offset 6AF4h

This register specifies the size of the MOT region. If a request address [38:24] ANDed with MOT_OUT_MASK[28:14] matches the MOT_OUT_BASE[38:24], then the request falls within the MOT_OUT region.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6AF4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Asset Classification AC[2]: GT Implicit Writeback Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback HITM data from GT to be returned to the requester. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester, depending on the setting of the IA_IWB_EN bit.
30	0h RW	**Asset Classification AC[1]: IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
29	0h RO	**Reserved (RESERVED_1):** Reserved
28:14	0h RW	**MOT Out Mask (MOT_OUT_MASK):** Specifies the size of the MOT region. If Request Address [38:24] ANDed with MOT_OUT_MASK[28:14] matches the MOT_OUT_BASE[28:14] then the request falls within the MOT_OUT region.
13:0	0h RO	**Reserved (RESERVED_0):** Reserved

3.390 MOT Buffer Control Policy (B_CR_BMOT_BUF_CP_0_0_0_MCHBAR) — Offset 6AF8h

This register controls the access policy to BMOT_BUF_RAC, BMOT_BUF_WAC, and, self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6AF8h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
62	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
61	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
60	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
59	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
58	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
57	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
56	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
55	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
54	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
53	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
52	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
51	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
50	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
49	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
48	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
47	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
46	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
45	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
44	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
43	1h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
42	1h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
41	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
40	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
39	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
38	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
37	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
36	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
35	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
34	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
33	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
32	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
31	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
30	1h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
29	1h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
28	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
27	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
26	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
25	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
24	1h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
23	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
22	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
20	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
19	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
18	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
17	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
16	1h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
15	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
14	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
13	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
12	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
11	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
10	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
9	1h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
8	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
7	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
6	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
5	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
4	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
3	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
1	1h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.
0	0h RW	**MOT Buffer Control Policy (MOT_BUF_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BMOT_BUF_RAC, BMOT_BUF_WAC and BMOT_BUF_CP registers, based on the value from each agent's 6bit SAI field.

3.391 MOT Buffer Read Access Policy (B_CR_BMOT_BUF_RAC_0_0_0_MCHBAR) — Offset 6B00h

This register, along with MOTBASE, MOTMASK and MOT_BUF_WAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to MOT. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6B00h	60010017 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**MOT Read Access Policy 63 (MOT_BUF_READ_POL_63):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
62	0h RO	**MOT Read Access Policy 62 (MOT_BUF_READ_POL_62):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
61	0h RO	**MOT Read Access Policy 61 (MOT_BUF_READ_POL_61):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
60	0h RO	**MOT Read Access Policy 60 (MOT_BUF_READ_POL_60):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
59	0h RW	**MOT Read Access Policy 59 (MOT_BUF_READ_POL_59):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
58	0h RO	**MOT Read Access Policy 58 (MOT_BUF_READ_POL_58):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
57	0h RO	**MOT Read Access Policy 57 (MOT_BUF_READ_POL_57):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
56	0h RW	**MOT Read Access Policy 56 (MOT_BUF_READ_POL_56):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
55	0h RW	**MOT Read Access Policy 55 (MOT_BUF_READ_POL_55):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
54	0h RW	**MOT Read Access Policy 54 (MOT_BUF_READ_POL_54):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
53	0h RO	**MOT Read Access Policy 53 (MOT_BUF_READ_POL_53):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
52	0h RO	**MOT Read Access Policy 52 (MOT_BUF_READ_POL_52):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
51	0h RO	**MOT Read Access Policy 51 (MOT_BUF_READ_POL_51):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
50	0h RW	**MOT Read Access Policy 50 (MOT_BUF_READ_POL_50):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
49	0h RW	**MOT Read Access Policy 49 (MOT_BUF_READ_POL_49):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
48	0h RW	**MOT Read Access Policy 48 (MOT_BUF_READ_POL_48):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
47	0h RO	**MOT Read Access Policy 47 (MOT_BUF_READ_POL_47):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
46	0h RO	**MOT Read Access Policy 46 (MOT_BUF_READ_POL_46):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
45	0h RO	**MOT Read Access Policy 45 (MOT_BUF_READ_POL_45):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
44	0h RW	**MOT Read Access Policy 44 (MOT_BUF_READ_POL_44):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
43	0h RW	**MOT Read Access Policy 43 (MOT_BUF_READ_POL_43):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
42	0h RW	**MOT Read Access Policy 42 (MOT_BUF_READ_POL_42):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
41	0h RW	**MOT Read Access Policy 41 (MOT_BUF_READ_POL_41):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
40	0h RW	**MOT Read Access Policy 40 (MOT_BUF_READ_POL_40):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
39	0h RO	**MOT Read Access Policy 39 (MOT_BUF_READ_POL_39):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
38	0h RW	**MOT Read Access Policy 38 (MOT_BUF_READ_POL_38):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
37	0h RO	**MOT Read Access Policy 37 (MOT_BUF_READ_POL_37):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
36	0h RW	**MOT Read Access Policy 36 (MOT_BUF_READ_POL_36):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
35	0h RO	**MOT Read Access Policy 35 (MOT_BUF_READ_POL_35):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
34	0h RW	**MOT Read Access Policy 34 (MOT_BUF_READ_POL_34):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
33	0h RW	**MOT Read Access Policy 33 (MOT_BUF_READ_POL_33):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
32	0h RW	**MOT Read Access Policy 32 (MOT_BUF_READ_POL_32):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
31	0h RO	**MOT Read Access Policy 31 (MOT_BUF_READ_POL_31):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
30	1h RW	**MOT Read Access Policy 30 (MOT_BUF_READ_POL_30):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
29	1h RW	**MOT Read Access Policy 29 (MOT_BUF_READ_POL_29):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
28	0h RW	**MOT Read Access Policy 28 (MOT_BUF_READ_POL_28):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
27	0h RW	**MOT Read Access Policy 27 (MOT_BUF_READ_POL_27):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
26	0h RW	**MOT Read Access Policy 26 (MOT_BUF_READ_POL_26):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
25	0h RW	**MOT Read Access Policy 25 (MOT_BUF_READ_POL_25):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
24	0h RW	**MOT Read Access Policy 24 (MOT_BUF_READ_POL_24):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
23	0h RO	**MOT Read Access Policy 23 (MOT_BUF_READ_POL_23):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
22	0h RO	**MOT Read Access Policy 22 (MOT_BUF_READ_POL_22):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
21	0h RW	**MOT Read Access Policy 21 (MOT_BUF_READ_POL_21):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
20	0h RO	**MOT Read Access Policy 20 (MOT_BUF_READ_POL_20):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
19	0h RO	**MOT Read Access Policy 19 (MOT_BUF_READ_POL_19):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
18	0h RO	**MOT Read Access Policy 18 (MOT_BUF_READ_POL_18):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
17	0h RW	**MOT Read Access Policy 17 (MOT_BUF_READ_POL_17):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
16	1h RW	**MOT Read Access Policy 16 (MOT_BUF_READ_POL_16):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
15	0h RO	**MOT Read Access Policy 15 (MOT_BUF_READ_POL_15):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
14	0h RO	**MOT Read Access Policy 14 (MOT_BUF_READ_POL_14):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
13	0h RW	**MOT Read Access Policy 13 (MOT_BUF_READ_POL_13):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
12	0h RW	**MOT Read Access Policy 12 (MOT_BUF_READ_POL_12):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
11	0h RO	**MOT Read Access Policy 11 (MOT_BUF_READ_POL_11):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
10	0h RO	**MOT Read Access Policy 10 (MOT_BUF_READ_POL_10):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
9	0h RO	**MOT Read Access Policy 9 (MOT_BUF_READ_POL_9):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
8	0h RW	**MOT Read Access Policy 8 (MOT_BUF_READ_POL_8):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
7	0h RW	**MOT Read Access Policy 7 (MOT_BUF_READ_POL_7):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
6	0h RW	**MOT Read Access Policy 6 (MOT_BUF_READ_POL_6):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
5	0h RW	**MOT Read Access Policy 5 (MOT_BUF_READ_POL_5):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
4	1h RW	**MOT Read Access Policy 4 (MOT_BUF_READ_POL_4):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
3	0h RW	**MOT Read Access Policy 3 (MOT_BUF_READ_POL_3):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
2	1h RW	**MOT Read Access Policy 2 (MOT_BUF_READ_POL_2):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
1	1h RW	**MOT Read Access Policy 1 (MOT_BUF_READ_POL_1):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.
0	1h RO	**MOT Read Access Policy 0 (MOT_BUF_READ_POL_0):** Bit vector used to determine which agents are allowed read access to the MOT region, based on each agent's 6bit encoded SAI value.

3.392 MOT Buffer Write Access Policy (B_CR_BMOT_BUF_WAC_0_0_0_MCHBAR) — Offset 6B08h

This register, along with MOTBASE, MOTMASK and MOT_BUF_RAC, defines an isolated region of memory that can be masked to prohibit certain agents from accessing memory. It is programmed with an SAI Policy that indicates which agents in the system are allowed to perform read operations to MOT. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6B08h	60010000 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**MOT Write Access Policy 63 (MOT_BUF_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
62	0h RO	**MOT Write Access Policy 62 (MOT_BUF_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
61	0h RO	**MOT Write Access Policy 61 (MOT_BUF_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
60	0h RO	**MOT Write Access Policy 60 (MOT_BUF_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
59	0h RW	**MOT Write Access Policy 59 (MOT_BUF_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
58	0h RO	**MOT Write Access Policy 58 (MOT_BUF_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
57	0h RO	**MOT Write Access Policy 57 (MOT_BUF_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
56	0h RW	**MOT Write Access Policy 56 (MOT_BUF_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
55	0h RW	**MOT Write Access Policy 55 (MOT_BUF_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
54	0h RW	**MOT Write Access Policy 54 (MOT_BUF_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
53	0h RO	**MOT Write Access Policy 53 (MOT_BUF_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
52	0h RO	**MOT Write Access Policy 52 (MOT_BUF_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
51	0h RO	**MOT Write Access Policy 51 (MOT_BUF_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
50	0h RW	**MOT Write Access Policy 50 (MOT_BUF_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
49	0h RW	**MOT Write Access Policy 49 (MOT_BUF_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
48	0h RW	**MOT Write Access Policy 48 (MOT_BUF_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
47	0h RO	**MOT Write Access Policy 47 (MOT_BUF_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
46	0h RO	**MOT Write Access Policy 46 (MOT_BUF_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
45	0h RO	**MOT Write Access Policy 45 (MOT_BUF_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
44	0h RW	**MOT Write Access Policy 44 (MOT_BUF_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
43	0h RW	**MOT Write Access Policy 43 (MOT_BUF_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
42	0h RW	**MOT Write Access Policy 42 (MOT_BUF_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
41	0h RW	**MOT Write Access Policy 41 (MOT_BUF_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
40	0h RW	**MOT Write Access Policy 40 (MOT_BUF_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
39	0h RO	**MOT Write Access Policy 39 (MOT_BUF_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
38	0h RW	**MOT Write Access Policy 38 (MOT_BUF_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
37	0h RO	**MOT Write Access Policy 37 (MOT_BUF_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
36	0h RW	**MOT Write Access Policy 36 (MOT_BUF_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
35	0h RO	**MOT Write Access Policy 35 (MOT_BUF_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
34	0h RW	**MOT Write Access Policy 34 (MOT_BUF_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
33	0h RW	**MOT Write Access Policy 33 (MOT_BUF_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
32	0h RW	**MOT Write Access Policy 32 (MOT_BUF_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**MOT Write Access Policy 31 (MOT_BUF_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
30	1h RW	**MOT Write Access Policy 30 (MOT_BUF_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
29	1h RW	**MOT Write Access Policy 29 (MOT_BUF_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
28	0h RW	**MOT Write Access Policy 28 (MOT_BUF_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
27	0h RW	**MOT Write Access Policy 27 (MOT_BUF_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
26	0h RW	**MOT Write Access Policy 26 (MOT_BUF_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
25	0h RW	**MOT Write Access Policy 25 (MOT_BUF_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
24	0h RW	**MOT Write Access Policy 24 (MOT_BUF_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
23	0h RO	**MOT Write Access Policy 23 (MOT_BUF_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
22	0h RO	**MOT Write Access Policy 22 (MOT_BUF_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
21	0h RW	**MOT Write Access Policy 21 (MOT_BUF_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
20	0h RO	**MOT Write Access Policy 20 (MOT_BUF_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
19	0h RO	**MOT Write Access Policy 19 (MOT_BUF_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
18	0h RO	**MOT Write Access Policy 18 (MOT_BUF_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
17	0h RW	**MOT Write Access Policy 17 (MOT_BUF_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
16	1h RW	**MOT Write Access Policy 16 (MOT_BUF_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
15	0h RO	**MOT Write Access Policy 15 (MOT_BUF_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
14	0h RO	**MOT Write Access Policy 14 (MOT_BUF_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
13	0h RW	**MOT Write Access Policy 13 (MOT_BUF_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**MOT Write Access Policy 12 (MOT_BUF_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
11	0h RO	**MOT Write Access Policy 11 (MOT_BUF_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
10	0h RO	**MOT Write Access Policy 10 (MOT_BUF_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
9	0h RO	**MOT Write Access Policy 9 (MOT_BUF_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
8	0h RW	**MOT Write Access Policy 8 (MOT_BUF_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
7	0h RW	**MOT Write Access Policy 7 (MOT_BUF_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
6	0h RW	**MOT Write Access Policy 6 (MOT_BUF_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
5	0h RW	**MOT Write Access Policy 5 (MOT_BUF_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
4	0h RW	**MOT Write Access Policy 4 (MOT_BUF_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
3	0h RW	**MOT Write Access Policy 3 (MOT_BUF_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
2	0h RW	**MOT Write Access Policy 2 (MOT_BUF_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
1	0h RW	**MOT Write Access Policy 1 (MOT_BUF_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.
0	0h RO	**MOT Write Access Policy 0 (MOT_BUF_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to the MOT region, based on each agent's 6bit encoded SAI value.

3.393 IMR Global BM Control Policy (B_CR_BIMRGLOBAL_BM_CP_0_0_0_MCHBAR) — Offset 6B10h

Defines the policy register that specifies who is allowed write access to BIMRGLOBAL_BM_WAC register.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6B10h	40001000202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**BM Control Policy (BM_CTRL_POL_63):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
62	0h RW	**BM Control Policy (BM_CTRL_POL_62):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
61	0h RW	**BM Control Policy (BM_CTRL_POL_61):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
60	0h RW	**BM Control Policy (BM_CTRL_POL_60):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
59	0h RW	**BM Control Policy (BM_CTRL_POL_59):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
58	0h RW	**BM Control Policy (BM_CTRL_POL_58):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
57	0h RW	**BM Control Policy (BM_CTRL_POL_57):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
56	0h RW	**BM Control Policy (BM_CTRL_POL_56):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
55	0h RW	**BM Control Policy (BM_CTRL_POL_55):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
54	0h RW	**BM Control Policy (BM_CTRL_POL_54):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
53	0h RW	**BM Control Policy (BM_CTRL_POL_53):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
52	0h RW	**BM Control Policy (BM_CTRL_POL_52):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
51	0h RW	**BM Control Policy (BM_CTRL_POL_51):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
50	0h RW	**BM Control Policy (BM_CTRL_POL_50):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
49	0h RW	**BM Control Policy (BM_CTRL_POL_49):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
48	0h RW	**BM Control Policy (BM_CTRL_POL_48):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
47	0h RW	**BM Control Policy (BM_CTRL_POL_47):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
46	0h RW	**BM Control Policy (BM_CTRL_POL_46):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
45	0h RW	**BM Control Policy (BM_CTRL_POL_45):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
44	0h RW	**BM Control Policy (BM_CTRL_POL_44):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
43	0h RW	**BM Control Policy (BM_CTRL_POL_43):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
42	1h RW	**BM Control Policy (BM_CTRL_POL_42):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
41	0h RW	**BM Control Policy (BM_CTRL_POL_41):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
40	0h RW	**BM Control Policy (BM_CTRL_POL_40):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
39	0h RW	**BM Control Policy (BM_CTRL_POL_39):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
38	0h RW	**BM Control Policy (BM_CTRL_POL_38):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
37	0h RW	**BM Control Policy (BM_CTRL_POL_37):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
36	0h RW	**BM Control Policy (BM_CTRL_POL_36):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
35	0h RW	**BM Control Policy (BM_CTRL_POL_35):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
34	0h RW	**BM Control Policy (BM_CTRL_POL_34):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
33	0h RW	**BM Control Policy (BM_CTRL_POL_33):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
32	0h RW	**BM Control Policy (BM_CTRL_POL_32):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
31	0h RW	**BM Control Policy (BM_CTRL_POL_31):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
30	0h RW	**BM Control Policy (BM_CTRL_POL_30):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
29	0h RW	**BM Control Policy (BM_CTRL_POL_29):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
28	0h RW	**BM Control Policy (BM_CTRL_POL_28):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
27	0h RW	**BM Control Policy (BM_CTRL_POL_27):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
26	0h RW	**BM Control Policy (BM_CTRL_POL_26):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**BM Control Policy (BM_CTRL_POL_25):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
24	1h RW	**BM Control Policy (BM_CTRL_POL_24):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
23	0h RW	**BM Control Policy (BM_CTRL_POL_23):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
22	0h RW	**BM Control Policy (BM_CTRL_POL_22):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
21	0h RW	**BM Control Policy (BM_CTRL_POL_21):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
20	0h RW	**BM Control Policy (BM_CTRL_POL_20):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
19	0h RW	**BM Control Policy (BM_CTRL_POL_19):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
18	0h RW	**BM Control Policy (BM_CTRL_POL_18):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
17	0h RW	**BM Control Policy (BM_CTRL_POL_17):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
16	0h RW	**BM Control Policy (BM_CTRL_POL_16):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
15	0h RW	**BM Control Policy (BM_CTRL_POL_15):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
14	0h RW	**BM Control Policy (BM_CTRL_POL_14):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
13	0h RW	**BM Control Policy (BM_CTRL_POL_13):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
12	0h RW	**BM Control Policy (BM_CTRL_POL_12):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
11	0h RW	**BM Control Policy (BM_CTRL_POL_11):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
10	0h RW	**BM Control Policy (BM_CTRL_POL_10):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
9	1h RW	**BM Control Policy (BM_CTRL_POL_9):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
8	0h RW	**BM Control Policy (BM_CTRL_POL_8):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
7	0h RW	**BM Control Policy (BM_CTRL_POL_7):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**BM Control Policy (BM_CTRL_POL_6):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
5	0h RW	**BM Control Policy (BM_CTRL_POL_5):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
4	0h RW	**BM Control Policy (BM_CTRL_POL_4):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
3	0h RW	**BM Control Policy (BM_CTRL_POL_3):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
2	0h RW	**BM Control Policy (BM_CTRL_POL_2):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
1	1h RW	**BM Control Policy (BM_CTRL_POL_1):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.
0	0h RW	**BM Control Policy (BM_CTRL_POL_0):** Bit vector used to determine which agents are allowed write access to BIMRGLOBAL_BM_WAC, based on each agent's 6bit encoded SAI value.

3.394 IMR Global BM Read Access Control (B_CR_BIMRGLOBAL_BM_RAC_0_0_0_MCHBAR) — Offset 6B18h

Defines the policy register that specifies who is allowed to read the BASE/MASK registers for IMR0-IMR15. This single common policy register protects reads to all base/mask registers.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6B18h	FFFFFFFFFFFFFFFF h

Bit Range	Default & Access	Field Name (ID): Description
63:0	FFFFFFFFFFFFFFFFh RO	**BM Read Access Policy (BM_READ_POL):** Bit vector used to determine which agents are allowed read access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.

3.395 IMR Global BM Write Access Policy (B_CR_BIMRGLOBAL_BM_WAC_0_0_0_MCHBAR) — Offset 6B20h

Defines the policy register that specifies who is allowed to overwrite the BASE/MASK registers for IMR0-IMR15. This single common policy register protects writes to all base/mask registers.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6B20h	40001000202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**BM Write Access Policy (BM_WRITE_POL_63):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
62	0h RW	**BM Write Access Policy (BM_WRITE_POL_62):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
61	0h RW	**BM Write Access Policy (BM_WRITE_POL_61):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
60	0h RW	**BM Write Access Policy (BM_WRITE_POL_60):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
59	0h RW	**BM Write Access Policy (BM_WRITE_POL_59):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
58	0h RW	**BM Write Access Policy (BM_WRITE_POL_58):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
57	0h RW	**BM Write Access Policy (BM_WRITE_POL_57):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
56	0h RW	**BM Write Access Policy (BM_WRITE_POL_56):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
55	0h RW	**BM Write Access Policy (BM_WRITE_POL_55):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
54	0h RW	**BM Write Access Policy (BM_WRITE_POL_54):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
53	0h RW	**BM Write Access Policy (BM_WRITE_POL_53):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
52	0h RW	**BM Write Access Policy (BM_WRITE_POL_52):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
51	0h RW	**BM Write Access Policy (BM_WRITE_POL_51):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
50	0h RW	**BM Write Access Policy (BM_WRITE_POL_50):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
49	0h RW	**BM Write Access Policy (BM_WRITE_POL_49):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
48	0h RW	**BM Write Access Policy (BM_WRITE_POL_48):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
47	0h RW	**BM Write Access Policy (BM_WRITE_POL_47):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
46	0h RW	**BM Write Access Policy (BM_WRITE_POL_46):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
45	0h RW	**BM Write Access Policy (BM_WRITE_POL_45):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
44	0h RW	**BM Write Access Policy (BM_WRITE_POL_44):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
43	0h RW	**BM Write Access Policy (BM_WRITE_POL_43):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
42	1h RW	**BM Write Access Policy (BM_WRITE_POL_42):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
41	0h RW	**BM Write Access Policy (BM_WRITE_POL_41):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
40	0h RW	**BM Write Access Policy (BM_WRITE_POL_40):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
39	0h RW	**BM Write Access Policy (BM_WRITE_POL_39):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
38	0h RW	**BM Write Access Policy (BM_WRITE_POL_38):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
37	0h RW	**BM Write Access Policy (BM_WRITE_POL_37):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
36	0h RW	**BM Write Access Policy (BM_WRITE_POL_36):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
35	0h RW	**BM Write Access Policy (BM_WRITE_POL_35):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
34	0h RW	**BM Write Access Policy (BM_WRITE_POL_34):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
33	0h RW	**BM Write Access Policy (BM_WRITE_POL_33):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
32	0h RW	**BM Write Access Policy (BM_WRITE_POL_32):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
31	0h RW	**BM Write Access Policy (BM_WRITE_POL_31):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
30	0h RW	**BM Write Access Policy (BM_WRITE_POL_30):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
29	0h RW	**BM Write Access Policy (BM_WRITE_POL_29):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
28	0h RW	**BM Write Access Policy (BM_WRITE_POL_28):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
27	0h RW	**BM Write Access Policy (BM_WRITE_POL_27):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
26	0h RW	**BM Write Access Policy (BM_WRITE_POL_26):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
25	0h RW	**BM Write Access Policy (BM_WRITE_POL_25):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
24	1h RW	**BM Write Access Policy (BM_WRITE_POL_24):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
23	0h RW	**BM Write Access Policy (BM_WRITE_POL_23):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
22	0h RW	**BM Write Access Policy (BM_WRITE_POL_22):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
21	0h RW	**BM Write Access Policy (BM_WRITE_POL_21):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
20	0h RW	**BM Write Access Policy (BM_WRITE_POL_20):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
19	0h RW	**BM Write Access Policy (BM_WRITE_POL_19):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
18	0h RW	**BM Write Access Policy (BM_WRITE_POL_18):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
17	0h RW	**BM Write Access Policy (BM_WRITE_POL_17):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
16	0h RW	**BM Write Access Policy (BM_WRITE_POL_16):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
15	0h RW	**BM Write Access Policy (BM_WRITE_POL_15):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
14	0h RW	**BM Write Access Policy (BM_WRITE_POL_14):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
13	0h RW	**BM Write Access Policy (BM_WRITE_POL_13):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
12	0h RW	**BM Write Access Policy (BM_WRITE_POL_12):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
11	0h RW	**BM Write Access Policy (BM_WRITE_POL_11):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
10	0h RW	**BM Write Access Policy (BM_WRITE_POL_10):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
9	1h RW	**BM Write Access Policy (BM_WRITE_POL_9):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
8	0h RW	**BM Write Access Policy (BM_WRITE_POL_8):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
7	0h RW	**BM Write Access Policy (BM_WRITE_POL_7):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
6	0h RW	**BM Write Access Policy (BM_WRITE_POL_6):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
5	0h RW	**BM Write Access Policy (BM_WRITE_POL_5):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
4	0h RW	**BM Write Access Policy (BM_WRITE_POL_4):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
3	0h RW	**BM Write Access Policy (BM_WRITE_POL_3):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
2	0h RW	**BM Write Access Policy (BM_WRITE_POL_2):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
1	1h RW	**BM Write Access Policy (BM_WRITE_POL_1):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.
0	0h RW	**BM Write Access Policy (BM_WRITE_POL_0):** Bit vector used to determine which agents are allowed write access to all IMRxBASE and IMRxMASK registers, based on each agent's 6bit encoded SAI value.

3.396 Graphics Stolen Memory Control Policy (B_CR_BGSMCP_0_0_0_MCHBAR) — Offset 6B28h

This register controls the access policy to the Read Access Policy BGSMRAC, Write Access Policy BGSMWAC and self-referentially, to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6B28h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**GSM Control Policy (GSM_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
62	0h RW	**GSM Control Policy (GSM_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
61	0h RW	**GSM Control Policy (GSM_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
60	0h RW	**GSM Control Policy (GSM_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
59	0h RW	**GSM Control Policy (GSM_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
58	0h RW	**GSM Control Policy (GSM_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
57	0h RW	**GSM Control Policy (GSM_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
56	0h RW	**GSM Control Policy (GSM_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
55	0h RW	**GSM Control Policy (GSM_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
54	0h RW	**GSM Control Policy (GSM_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
53	0h RW	**GSM Control Policy (GSM_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
52	0h RW	**GSM Control Policy (GSM_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
51	0h RW	**GSM Control Policy (GSM_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
50	0h RW	**GSM Control Policy (GSM_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
49	0h RW	**GSM Control Policy (GSM_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
48	0h RW	**GSM Control Policy (GSM_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
47	0h RW	**GSM Control Policy (GSM_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
46	0h RW	**GSM Control Policy (GSM_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
45	0h RW	**GSM Control Policy (GSM_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
44	0h RW	**GSM Control Policy (GSM_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
43	1h RW	**GSM Control Policy (GSM_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
42	1h RW	**GSM Control Policy (GSM_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
41	0h RW	**GSM Control Policy (GSM_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
40	0h RW	**GSM Control Policy (GSM_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
39	0h RW	**GSM Control Policy (GSM_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
38	0h RW	**GSM Control Policy (GSM_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
37	0h RW	**GSM Control Policy (GSM_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
36	0h RW	**GSM Control Policy (GSM_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
35	0h RW	**GSM Control Policy (GSM_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
34	0h RW	**GSM Control Policy (GSM_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
33	0h RW	**GSM Control Policy (GSM_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
32	0h RW	**GSM Control Policy (GSM_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
31	0h RW	**GSM Control Policy (GSM_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
30	1h RW	**GSM Control Policy (GSM_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
29	1h RW	**GSM Control Policy (GSM_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
28	0h RW	**GSM Control Policy (GSM_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
27	0h RW	**GSM Control Policy (GSM_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
26	0h RW	**GSM Control Policy (GSM_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
25	0h RW	**GSM Control Policy (GSM_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
24	1h RW	**GSM Control Policy (GSM_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**GSM Control Policy (GSM_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
22	0h RW	**GSM Control Policy (GSM_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
21	0h RW	**GSM Control Policy (GSM_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
20	0h RW	**GSM Control Policy (GSM_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
19	0h RW	**GSM Control Policy (GSM_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
18	0h RW	**GSM Control Policy (GSM_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
17	0h RW	**GSM Control Policy (GSM_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
16	1h RW	**GSM Control Policy (GSM_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
15	0h RW	**GSM Control Policy (GSM_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
14	0h RW	**GSM Control Policy (GSM_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
13	0h RW	**GSM Control Policy (GSM_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
12	0h RW	**GSM Control Policy (GSM_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
11	0h RW	**GSM Control Policy (GSM_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
10	0h RW	**GSM Control Policy (GSM_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
9	1h RW	**GSM Control Policy (GSM_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
8	0h RW	**GSM Control Policy (GSM_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
7	0h RW	**GSM Control Policy (GSM_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
6	0h RW	**GSM Control Policy (GSM_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
5	0h RW	**GSM Control Policy (GSM_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW	**GSM Control Policy (GSM_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
3	0h RW	**GSM Control Policy (GSM_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
2	0h RW	**GSM Control Policy (GSM_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
1	1h RW	**GSM Control Policy (GSM_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.
0	0h RW	**GSM Control Policy (GSM_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the BGSMRAC, BGSMWAC and BGSMCP registers, based on the value from each agent's 6bit SAI field.

3.397 GSM Read Access Policy (B_CR_BGSMRAC_0_0_0_MCHBAR) — Offset 6B30h

This register configures the Read Access Policy to the memory range from BGSM to TOLUD1. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6B30h	1000600101 00 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**SMM Read Access Policy 63 (GSM_SAI_POL_63):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
62	0h RO	**SMM Read Access Policy 62 (GSM_SAI_POL_62):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
61	0h RO	**SMM Read Access Policy 61 (GSM_SAI_POL_61):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
60	0h RO	**SMM Read Access Policy 60 (GSM_SAI_POL_60):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
59	0h RW	**SMM Read Access Policy 59 (GSM_SAI_POL_59):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
58	0h RO	**SMM Read Access Policy 58 (GSM_SAI_POL_58):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
57	0h RO	**SMM Read Access Policy 57 (GSM_SAI_POL_57):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
56	0h RW	**SMM Read Access Policy 56 (GSM_SAI_POL_56):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
55	0h RW	**SMM Read Access Policy 55 (GSM_SAI_POL_55):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
54	0h RW	**SMM Read Access Policy 54 (GSM_SAI_POL_54):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
53	0h RO	**SMM Read Access Policy 53 (GSM_SAI_POL_53):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
52	0h RO	**SMM Read Access Policy 52 (GSM_SAI_POL_52):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
51	0h RO	**SMM Read Access Policy 51 (GSM_SAI_POL_51):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
50	0h RW	**SMM Read Access Policy 50 (GSM_SAI_POL_50):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
49	0h RW	**SMM Read Access Policy 49 (GSM_SAI_POL_49):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
48	0h RW	**SMM Read Access Policy 48 (GSM_SAI_POL_48):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
47	0h RO	**SMM Read Access Policy 47 (GSM_SAI_POL_47):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
46	0h RO	**SMM Read Access Policy 46 (GSM_SAI_POL_46):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
45	0h RO	**SMM Read Access Policy 45 (GSM_SAI_POL_45):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
44	1h RW	**SMM Read Access Policy 44 (GSM_SAI_POL_44):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
43	0h RW	**SMM Read Access Policy 43 (GSM_SAI_POL_43):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
42	0h RW	**SMM Read Access Policy 42 (GSM_SAI_POL_42):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
41	0h RW	**SMM Read Access Policy 41 (GSM_SAI_POL_41):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
40	0h RW	**SMM Read Access Policy 40 (GSM_SAI_POL_40):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
39	0h RO	**SMM Read Access Policy 39 (GSM_SAI_POL_39):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
38	0h RW	**SMM Read Access Policy 38 (GSM_SAI_POL_38):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
37	0h RO	**SMM Read Access Policy 37 (GSM_SAI_POL_37):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
36	0h RW	**SMM Read Access Policy 36 (GSM_SAI_POL_36):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
35	0h RO	**SMM Read Access Policy 35 (GSM_SAI_POL_35):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
34	0h RW	**SMM Read Access Policy 34 (GSM_SAI_POL_34):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
33	0h RW	**SMM Read Access Policy 33 (GSM_SAI_POL_33):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
32	0h RW	**SMM Read Access Policy 32 (GSM_SAI_POL_32):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
31	0h RO	**SMM Read Access Policy 31 (GSM_SAI_POL_31):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
30	1h RW	**SMM Read Access Policy 30 (GSM_SAI_POL_30):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
29	1h RW	**SMM Read Access Policy 29 (GSM_SAI_POL_29):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
28	0h RW	**SMM Read Access Policy 28 (GSM_SAI_POL_28):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
27	0h RW	**SMM Read Access Policy 27 (GSM_SAI_POL_27):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
26	0h RW	**SMM Read Access Policy 26 (GSM_SAI_POL_26):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
25	0h RW	**SMM Read Access Policy 25 (GSM_SAI_POL_25):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
24	0h RW	**SMM Read Access Policy 24 (GSM_SAI_POL_24):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
23	0h RO	**SMM Read Access Policy 23 (GSM_SAI_POL_23):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
22	0h RO	**SMM Read Access Policy 22 (GSM_SAI_POL_22):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
21	0h RW	**SMM Read Access Policy 21 (GSM_SAI_POL_21):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
20	0h RO	**SMM Read Access Policy 20 (GSM_SAI_POL_20):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
19	0h RO	**SMM Read Access Policy 19 (GSM_SAI_POL_19):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**SMM Read Access Policy 18 (GSM_SAI_POL_18):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
17	0h RW	**SMM Read Access Policy 17 (GSM_SAI_POL_17):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
16	1h RW	**SMM Read Access Policy 16 (GSM_SAI_POL_16):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
15	0h RO	**SMM Read Access Policy 15 (GSM_SAI_POL_15):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
14	0h RO	**SMM Read Access Policy 14 (GSM_SAI_POL_14):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
13	0h RW	**SMM Read Access Policy 13 (GSM_SAI_POL_13):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
12	0h RW	**SMM Read Access Policy 12 (GSM_SAI_POL_12):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
11	0h RO	**SMM Read Access Policy 11 (GSM_SAI_POL_11):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
10	0h RO	**SMM Read Access Policy 10 (GSM_SAI_POL_10):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
9	0h RO	**SMM Read Access Policy 9 (GSM_SAI_POL_9):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
8	1h RW	**SMM Read Access Policy 8 (GSM_SAI_POL_8):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
7	0h RW	**SMM Read Access Policy 7 (GSM_SAI_POL_7):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
6	0h RW	**SMM Read Access Policy 6 (GSM_SAI_POL_6):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
5	0h RW	**SMM Read Access Policy 5 (GSM_SAI_POL_5):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
4	0h RW	**SMM Read Access Policy 4 (GSM_SAI_POL_4):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
3	0h RW	**SMM Read Access Policy 3 (GSM_SAI_POL_3):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
2	0h RW	**SMM Read Access Policy 2 (GSM_SAI_POL_2):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
1	0h RW	**SMM Read Access Policy 1 (GSM_SAI_POL_1):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
0	0h RO	**SMM Read Access Policy 0 (GSM_SAI_POL_0):** Bit vector used to determine which agents are allowed read access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.

3.398 GSM Write Access Policy (B_CR_BGSMWAC_0_0_0_MCHBAR) — Offset 6B38h

This register configures the Write Access Policy for the memory range from BGSM to TOLUD1. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6B38h	100060010100 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**GSM Write Access Policy 63 (GSM_SAI_POL_63):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
62	0h RO	**GSM Write Access Policy 62 (GSM_SAI_POL_62):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
61	0h RO	**GSM Write Access Policy 61 (GSM_SAI_POL_61):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
60	0h RO	**GSM Write Access Policy 60 (GSM_SAI_POL_60):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
59	0h RW	**GSM Write Access Policy 59 (GSM_SAI_POL_59):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
58	0h RO	**GSM Write Access Policy 58 (GSM_SAI_POL_58):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
57	0h RO	**GSM Write Access Policy 57 (GSM_SAI_POL_57):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
56	0h RW	**GSM Write Access Policy 56 (GSM_SAI_POL_56):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
55	0h RW	**GSM Write Access Policy 55 (GSM_SAI_POL_55):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
54	0h RW	**GSM Write Access Policy 54 (GSM_SAI_POL_54):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
53	0h RO	**GSM Write Access Policy 53 (GSM_SAI_POL_53):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
52	0h RO	**GSM Write Access Policy 52 (GSM_SAI_POL_52):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
51	0h RO	**GSM Write Access Policy 51 (GSM_SAI_POL_51):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
50	0h RW	**GSM Write Access Policy 50 (GSM_SAI_POL_50):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
49	0h RW	**GSM Write Access Policy 49 (GSM_SAI_POL_49):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
48	0h RW	**GSM Write Access Policy 48 (GSM_SAI_POL_48):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
47	0h RO	**GSM Write Access Policy 47 (GSM_SAI_POL_47):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
46	0h RO	**GSM Write Access Policy 46 (GSM_SAI_POL_46):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
45	0h RO	**GSM Write Access Policy 45 (GSM_SAI_POL_45):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
44	1h RW	**GSM Write Access Policy 44 (GSM_SAI_POL_44):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
43	0h RW	**GSM Write Access Policy 43 (GSM_SAI_POL_43):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
42	0h RW	**GSM Write Access Policy 42 (GSM_SAI_POL_42):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
41	0h RW	**GSM Write Access Policy 41 (GSM_SAI_POL_41):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
40	0h RW	**GSM Write Access Policy 40 (GSM_SAI_POL_40):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
39	0h RO	**GSM Write Access Policy 39 (GSM_SAI_POL_39):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
38	0h RW	**GSM Write Access Policy 38 (GSM_SAI_POL_38):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
37	0h RO	**GSM Write Access Policy 37 (GSM_SAI_POL_37):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
36	0h RW	**GSM Write Access Policy 36 (GSM_SAI_POL_36):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
35	0h RO	**GSM Write Access Policy 35 (GSM_SAI_POL_35):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
34	0h RW	**GSM Write Access Policy 34 (GSM_SAI_POL_34):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
33	0h RW	**GSM Write Access Policy 33 (GSM_SAI_POL_33):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
32	0h RW	**GSM Write Access Policy 32 (GSM_SAI_POL_32):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
31	0h RO	**GSM Write Access Policy 31 (GSM_SAI_POL_31):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
30	1h RW	**GSM Write Access Policy 30 (GSM_SAI_POL_30):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
29	1h RW	**GSM Write Access Policy 29 (GSM_SAI_POL_29):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
28	0h RW	**GSM Write Access Policy 28 (GSM_SAI_POL_28):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
27	0h RW	**GSM Write Access Policy 27 (GSM_SAI_POL_27):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
26	0h RW	**GSM Write Access Policy 26 (GSM_SAI_POL_26):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
25	0h RW	**GSM Write Access Policy 25 (GSM_SAI_POL_25):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
24	0h RW	**GSM Write Access Policy 24 (GSM_SAI_POL_24):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
23	0h RO	**GSM Write Access Policy 23 (GSM_SAI_POL_23):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
22	0h RO	**GSM Write Access Policy 22 (GSM_SAI_POL_22):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
21	0h RW	**GSM Write Access Policy 21 (GSM_SAI_POL_21):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
20	0h RO	**GSM Write Access Policy 20 (GSM_SAI_POL_20):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
19	0h RO	**GSM Write Access Policy 19 (GSM_SAI_POL_19):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
18	0h RO	**GSM Write Access Policy 18 (GSM_SAI_POL_18):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
17	0h RW	**GSM Write Access Policy 17 (GSM_SAI_POL_17):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
16	1h RW	**GSM Write Access Policy 16 (GSM_SAI_POL_16):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
15	0h RO	**GSM Write Access Policy 15 (GSM_SAI_POL_15):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
14	0h RO	**GSM Write Access Policy 14 (GSM_SAI_POL_14):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
13	0h RW	**GSM Write Access Policy 13 (GSM_SAI_POL_13):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
12	0h RW	**GSM Write Access Policy 12 (GSM_SAI_POL_12):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RO	**GSM Write Access Policy 11 (GSM_SAI_POL_11):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
10	0h RO	**GSM Write Access Policy 10 (GSM_SAI_POL_10):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
9	0h RO	**GSM Write Access Policy 9 (GSM_SAI_POL_9):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
8	1h RW	**GSM Write Access Policy 8 (GSM_SAI_POL_8):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
7	0h RW	**GSM Write Access Policy 7 (GSM_SAI_POL_7):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
6	0h RW	**GSM Write Access Policy 6 (GSM_SAI_POL_6):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
5	0h RW	**GSM Write Access Policy 5 (GSM_SAI_POL_5):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
4	0h RW	**GSM Write Access Policy 4 (GSM_SAI_POL_4):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
3	0h RW	**GSM Write Access Policy 3 (GSM_SAI_POL_3):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
2	0h RW	**GSM Write Access Policy 2 (GSM_SAI_POL_2):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
1	0h RW	**GSM Write Access Policy 1 (GSM_SAI_POL_1):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.
0	0h RO	**GSM Write Access Policy 0 (GSM_SAI_POL_0):** Bit vector used to determine which agents are allowed write access to the memory range from BGSM to TOLUD1, based on each agent's 6bit encoded SAI value.

3.399 TPM Control Policy (B_CR_TPM_CP_0_0_0_MCHBAR) — Offset 6B40h

Defines the control policy register for TPM group.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6B40h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**TPM Control Policy (CTRL_POL_63):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
62	0h RW	**TPM Control Policy (CTRL_POL_62):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
61	0h RW	**TPM Control Policy (CTRL_POL_61):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
60	0h RW	**TPM Control Policy (CTRL_POL_60):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
59	0h RW	**TPM Control Policy (CTRL_POL_59):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
58	0h RW	**TPM Control Policy (CTRL_POL_58):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
57	0h RW	**TPM Control Policy (CTRL_POL_57):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
56	0h RW	**TPM Control Policy (CTRL_POL_56):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
55	0h RW	**TPM Control Policy (CTRL_POL_55):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
54	0h RW	**TPM Control Policy (CTRL_POL_54):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
53	0h RW	**TPM Control Policy (CTRL_POL_53):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
52	0h RW	**TPM Control Policy (CTRL_POL_52):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
51	0h RW	**TPM Control Policy (CTRL_POL_51):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
50	0h RW	**TPM Control Policy (CTRL_POL_50):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
49	0h RW	**TPM Control Policy (CTRL_POL_49):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
48	0h RW	**TPM Control Policy (CTRL_POL_48):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
47	0h RW	**TPM Control Policy (CTRL_POL_47):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
46	0h RW	**TPM Control Policy (CTRL_POL_46):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
45	0h RW	**TPM Control Policy (CTRL_POL_45):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
44	0h RW	**TPM Control Policy (CTRL_POL_44):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
43	1h RW	**TPM Control Policy (CTRL_POL_43):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
42	1h RW	**TPM Control Policy (CTRL_POL_42):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
41	0h RW	**TPM Control Policy (CTRL_POL_41):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
40	0h RW	**TPM Control Policy (CTRL_POL_40):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
39	0h RW	**TPM Control Policy (CTRL_POL_39):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
38	0h RW	**TPM Control Policy (CTRL_POL_38):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
37	0h RW	**TPM Control Policy (CTRL_POL_37):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
36	0h RW	**TPM Control Policy (CTRL_POL_36):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
35	0h RW	**TPM Control Policy (CTRL_POL_35):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
34	0h RW	**TPM Control Policy (CTRL_POL_34):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
33	0h RW	**TPM Control Policy (CTRL_POL_33):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
32	0h RW	**TPM Control Policy (CTRL_POL_32):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
31	0h RW	**TPM Control Policy (CTRL_POL_31):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
30	1h RW	**TPM Control Policy (CTRL_POL_30):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
29	1h RW	**TPM Control Policy (CTRL_POL_29):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
28	0h RW	**TPM Control Policy (CTRL_POL_28):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
27	0h RW	**TPM Control Policy (CTRL_POL_27):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
26	0h RW	**TPM Control Policy (CTRL_POL_26):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
25	0h RW	**TPM Control Policy (CTRL_POL_25):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
24	1h RW	**TPM Control Policy (CTRL_POL_24):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
23	0h RW	**TPM Control Policy (CTRL_POL_23):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
22	0h RW	**TPM Control Policy (CTRL_POL_22):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
21	0h RW	**TPM Control Policy (CTRL_POL_21):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
20	0h RW	**TPM Control Policy (CTRL_POL_20):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
19	0h RW	**TPM Control Policy (CTRL_POL_19):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
18	0h RW	**TPM Control Policy (CTRL_POL_18):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
17	0h RW	**TPM Control Policy (CTRL_POL_17):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
16	1h RW	**TPM Control Policy (CTRL_POL_16):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
15	0h RW	**TPM Control Policy (CTRL_POL_15):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
14	0h RW	**TPM Control Policy (CTRL_POL_14):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
13	0h RW	**TPM Control Policy (CTRL_POL_13):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
12	0h RW	**TPM Control Policy (CTRL_POL_12):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
11	0h RW	**TPM Control Policy (CTRL_POL_11):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
10	0h RW	**TPM Control Policy (CTRL_POL_10):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
9	1h RW	**TPM Control Policy (CTRL_POL_9):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
8	0h RW	**TPM Control Policy (CTRL_POL_8):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
7	0h RW	**TPM Control Policy (CTRL_POL_7):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
6	0h RW	**TPM Control Policy (CTRL_POL_6):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**TPM Control Policy (CTRL_POL_5):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
4	0h RW	**TPM Control Policy (CTRL_POL_4):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
3	0h RW	**TPM Control Policy (CTRL_POL_3):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
2	0h RW	**TPM Control Policy (CTRL_POL_2):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
1	1h RW	**TPM Control Policy (CTRL_POL_1):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.
0	0h RW	**TPM Control Policy (CTRL_POL_0):** Bit vector used to determine which agents are allowed write access to TPM_AC register, based on each agent's 6-bit encoded SAI value.

3.400 TPM Access Control (B_CR_TPM_AC_0_0_0_MCHBAR) — Offset 6B48h

Defines the policy register that specifies who is allowed read/write access to the TPM_SELECTOR register.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6B48h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	C0061010202h RW	**TPM Access Policy (AC_POL):** Bit vector used to determine which agents are allowed read/write access to TPM_SELECTOR, based on each agent's 6-bit encoded SAI value.

3.401 BGSM Control Register (B_CR_BGSM_CTRL_0_0_0_MCHBAR) — Offset 6B50h

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6B50h	19 h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RO	**Reserved (RESERVED_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
4	1h RW	**RS0 Asset Classification Bit for Graphics and Data Stolen Memory (RS0_EN):** PII transactions from RS0 that hit the Graphics and Data Stolen Memory range will be allowed access only when both of the following conditions are met: a) Request SAI is in the legal permitted list as specified in the RAC/WAC policy registers and b) GSM_RS0_EN bit is set to 1. PII RS0 transactions targeting DRAM that do not hit any enabled IMR or special protected regions will always be allowed access.
3	1h RW	**GT Implicit WB Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback data HITM data from GT to be returned to the requester. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester depending on the setting of the IA_IWB_EN bit.
2	0h RW	**IA Implicit WB Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback data HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
1	0h RW	**Trace Enable (TRACE_EN):** Enables snooping of transactions to the Graphics Stolen Memory region by tracing agents.
0	1h RW	**Range Check Enable (RANGE_CHECK_EN):** Enables SAI checking for the memory range from BGSM to TOLUD-1

3.402 SMM Control Register (B_CR_BSMR_CTRL_0_0_0_MCHBAR) — Offset 6B54h

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6B54h	7 h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RO	**Reserved (RESERVED_0):** Reserved
4	0h RO	**RS0 Asset Classification Bit for SMM Region (RS0_EN):** No PII transaction is allowed access to SMM region. As such RS0 asset classification bit does not apply and hence provisioned as a readonly bit and set to 0.
3	0h RW	**GT Implicit Writeback Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback data HITM data from GT to be returned to the requester. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester, depending on the setting of the IA_IWB_EN bit.
2	1h RW	**IA Implicit Writeback Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback data HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester depending on the setting of the GT_IWB_EN bit.
1	1h RW	**Trace Enable (TRACE_EN):** Enables snooping of transactions to the SMM region by tracing agents.
0	1h RW	**Range Check Enable (RANGE_CHECK_EN):** Enables SAI checking for the SMM memory range: TSEGMB to BGSM.

3.403 Default VTd Control Register (B_CR_BDEFVTDPMR_CTRL_0_0_0_MCHBAR) — Offset 6B58h

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6B58h	1C h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RO	**Reserved (RESERVED_1):** Reserved
4	1h RW	**RS0 Asset Classification Bit for the High and Low VTd PMRs (RS0_EN):** PII transactions from RS0 that hit VTd PMR high and low memory ranges will be allowed access only when both of the following conditions are met: a) Request SAI is in the legal permitted list as specified in the RAC/WAC policy registers and b) VTDPMR_RS0_EN bit is set to 1. PII RS0 transactions targeting DRAM that do not hit any enabled IMR or special protected regions will always be allowed access.
3	1h RW	**GT Implicit Writeback Enable (GT_IWB_EN):** Enables implicit writebacks to protected region from GT caching agent. When set to 1, enables implicit writeback data HITM data from GT to be returned to the requester. When set to 0, inhibits HITM data from GT from being returned to the requester. HITM data from IA cores may be returned to the requester, depending on the setting of the IA_IWB_EN bit.
2	1h RW	**IA Implicit Writeback Enable (IA_IWB_EN):** Enables implicit writebacks to protected region from IA caching agent. When set to 1, enables implicit writeback data HITM data from IA cores to be returned to the requester. When set to 0, inhibits HITM data from IA cores from being returned to the requester. HITM data from GT may be returned to the requester, depending on the setting of the GT_IWB_EN bit.
1	0h RW	**Trace Enable (TRACE_EN):** Enables snooping of transactions to the Default VTd PHM and PLM regions by tracing agents.
0	0h RO	**Reserved (RESERVED_0):** Reserved

3.404 MOT Trigger Trace Control (B_CR_MOT_TRIG_TRACE_CTRL_0_0_0_MCHBAR) — Offset 6B7Ch

This register is the global MOT enable.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6B7Ch	E0000000 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	Eh RW	**Flush Timer Period (FLUSH_TIMER_PERIOD):** Specifies the duration of the periodic MOT flush timer. Duration is 2^N cycles of a 19.2 MHz clock, where N is the value specified in the field.
27	0h RO	**Reserved (RESERVED_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW/V	**Sticky Trigger 1 Match (STICKY_TRIGGER_1_MATCH):** 1 indicates Trigger 1 match. Cleared by powergood or explicit SW write.
25	0h RW/V	**Sticky Trigger 0 Match (STICKY_TRIGGER_0_MATCH):** 1 indicates Trigger 0 match. Cleared by powergood or explicit SW write.
24	0h RW/V	**Sticky Filter 1 Match (STICKY_FILTER_1_MATCH):** 1 indicates Filter 1 match. Cleared by powergood or explicit SW write.
23	0h RW/V	**Sticky Filter 0 Match (STICKY_FILTER_0_MATCH):** 1 indicates Filter 0 match. Cleared by powergood or explicit SW write.
22	0h RW/V	**MOT Storage Active (MOT_STORAGE_ACTIVE):** Current status of MOT storage to memory. Used to save/restore trigger state across power states.
21:16	0h RW	**MOT Trace Storage Stop Source (MOT_TRACE_STORAGE_STOP_SOURCE):** A bit field allowing multiple triggers to result in a trace storage stop. Storage stop is the logical OR of trigger sources selected. Note that selecting the same trigger to start and stop trace storage results in start trace storage only. • Bit position 5: External Trigger in 1 • Bit position 4: External Trigger in 0 • Bit position 3: MOT Trigger 1 • Bit position 2: MOT Trigger 0 • Bit position 1: MOT Filter 1 match • Bit position 0: MOT Filter 0 match
15:10	0h RW	**MOT Trace Storage Start Source (MOT_TRACE_STORAGE_START_SOURCE):** A bit field allowing multiple triggers to result in a trace storage start. Storage stop is the logical OR of trigger sources selected. Note that selecting the same trigger to start and stop trace storage results in start trace storage only. For bit positions, see description for MOT_TRACE_STORAGE_STOP_SOURCE field.
9	0h RW	**Enable External Trigger in 1 (ENABLE_EXTERNAL_TRIGGER_IN_1):** • 1 to enable external trigger 1 to MOT from rdu_mid • 0 to disable external trigger 1 to MOT from rdu_mid
8	0h RW	**Enable External Trigger in 0 (ENABLE_EXTERNAL_TRIGGER_IN_0):** • 1 to enable external trigger 0 to MOT from rdu_mid • 0 to disable external trigger 0 to MOT from rdu_mid
7:6	0h RW	**Trigger Out Source (TRIGGER_OUT_SOURCE):** Select source of trigger sent to rdu_mid. • 00b: Filter 0 • 01b: Filter 1 • 10b: Trigger 0 • 11b: Trigger 1
5	0h RW	**Enable External Trigger Out 0 (ENABLE_EXTERNAL_TRIGGER_OUT_0):** • 1: enable external trigger out to from MOT to rdu_mid • 0: disable external trigger out to from MOT to rdu_mid
4	0h RW	**Enable MOT Trigger 1 (ENABLE_MOT_TRIGGER_1):** • 1: enable MOT trigger 1 • 0: disable MOT trigger 1
3	0h RW	**Enable MOT Trigger 0 (ENABLE_MOT_TRIGGER_0):** • 1: enable MOT trigger 0 • 0: disable MOT trigger 0
2	0h RW	**Enable MOT Filter 1 (ENABLE_MOT_FILTER_1):** • 1: enable MOT filter 1 • 0: disable MOT filter 1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Enable MOT Filter 0 (ENABLE_MOT_FILTER_0):** • 1: enable MOT filter 0 • 0: disable MOT filter 0
0	0h RW	**Global MOT Enable (ENABLE_MOT):** • 1: enable MOT • 0: disable MOT

3.405 MOT Slice 0 Memory Pointer (B_CR_MOT_SLICE_0_MEM_PTR_0_0_0_MCHBAR) — Offset 6B80h

This register contains the memory address pointer used to determine where the MOT slice 0 cache lines are stored. On a write, the B-Unit logic will set itself to the value of the write Extraction of memory contents is performed as reads. On a read, the data returned should match the current value of the pointer reflecting incrementing updates from cache lines that have been stored. It also pins MOT to near or far memory.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6B80h	2 h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_0):** Reserved
38:6	0h RW/V	**MOT Memory Pointer (MOT_MEMORY_POINTER):** The current pointer into MOT slice 0 region. Cache line granularity bits [38:6].
5:2	0h RO	**Reserved (RESERVED_1):** Reserved
1	1h RW	**Near or Far Memory (NEAR_OR_FAR_MEMORY):** Pin MOT slice 0 to near1 or far0 memory. This logic is implemented in the 2LM. Thus, this register bit is shadowed in its entirety in the 2LM. Unused by MOT.
0	0h RW/V	**MOT Buffer Wrap (MOT_BUFFER_WRAP):** Indication that the MOT slice 0 buffer has wrapped since the last clear of this write cleared by power good or explicit SW write.

3.406 MOT Slice 1 Memory Pointer (B_CR_MOT_SLICE_1_MEM_PTR_0_0_0_MCHBAR) — Offset 6B88h

This register contains the memory address pointer used to determine where the MOT slice 1 cache lines are stored. On a write, the B-Unit logic will set itself to the value of the write. Extraction of memory contents are performed as reads. On a read, the data returned should match the current value of the pointer, reflecting incrementing updates from cache lines that have been stored. It also pins MOT to near or far memory.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6B88h	2 h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_0):** Reserved
38:6	0h RW/V	**MOT Memory Pointer (MOT_MEMORY_POINTER):** The current pointer into MOT slice 1 region. Cache line granularity bits [38:6].
5:2	0h RO	**Reserved (RESERVED_1):** Reserved
1	1h RW	**Near or Far Memory (NEAR_OR_FAR_MEMORY):** Pin MOT slice 1 to near1 or far0 memory. This logic is implemented in the 2LM. Thus, this register is shadowed in its entirety in the 2LM.
0	0h RW/V	**MOT Buffer Wrap (MOT_BUFFER_WRAP):** Indication that the MOT slice 1 buffer has wrapped since the last clear of this write cleared by power good or explicit SW write.

3.407 MOT Slice 0 Record ID (B_CR_MOT_SLICE_0_RECORD_ID_0_0_0_MCHBAR) — Offset 6B90h

This register contains the unique MOT record ID and start trace indication for slice 0. This state needs to be stored in a CR to allow it to persist across autoPG and s0ix transitions.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6B90h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	0h RO	**Reserved (RESERVED_0):** Reserved
26	0h RW/V	**First Packet Captured (FIRST_PKT_CAPTURED):** Indicates that MOT has captured the first data packet after tracing started.
25:0	0h RW/V	**Record ID (RECORD_ID):** The current MOT unique record ID number.

3.408 MOT Slice 1 Record ID (B_CR_MOT_SLICE_1_RECORD_ID_0_0_0_MCHBAR) — Offset 6B94h

This register contains the unique MOT record ID and start trace indication for slice 0. This state needs to be stored in a CR to allow it to persist across autoPG and s0ix transitions.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6B94h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	0h RO	**Reserved (RESERVED_0):** Reserved
26	0h RW/V	**First Packet Captured (FIRST_PKT_CAPTURED):** Indicates that MOT has captured the first data packet after tracing started.
25:0	0h RW/V	**Record ID (RECORD_ID):** The current MOT unique record ID number.

3.409 MOT Filter Match 0 (B_CR_MOT_FILTER_MATCH0_0_0_0_MCHBAR) — Offset 6BA0h

This register contains the value of the MOT 0 access filter Match address.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6BA0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_0):** Reserved
38:5	0h RW	**MOT 0 Filter Match Address (MOT_0_FILTER_MATCH_ADDRESS):** Access address to match. Match results in trace of access. Half cache line granularity bits 38:5 to support legacy devices.
4:0	0h RO	**Reserved (RESERVED_1):** Reserved

3.410 MOT Filter Mask (B_CR_MOT_FILTER_MASK0_0_0_0_MCHBAR) — Offset 6BA8h

This register contains the value of the MOT 0 access filter Mask address.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6BA8h	7FFFFFFFE0 h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_0):** Reserved
38:5	3FFFFFFFFh RW	**MOT 0 Filter Mask Address (MOT_0_FILTER_MASK_ADDRESS):** Access address bits to mask. A value of 1 ignores the corresponding address bit. Half cache line granularity bits 38:5 to support legacy devices.
4:0	0h RO	**Reserved (RESERVED_1):** Reserved

3.411 MOT Filter Match 1 (B_CR_MOT_FILTER_MATCH1_0_0_0_MCHBAR) — Offset 6BB0h

This register contains the value of the MOT 1 access filter Match address.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6BB0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_0):** Reserved
38:5	0h RW	**MOT 1 Filter Match Address (MOT_1_FILTER_MATCH_ADDRESS):** Access address to match. Match results in trace of access. Half cache line granularity bits 38:5 to support legacy devices.
4:0	0h RO	**Reserved (RESERVED_1):** Reserved

3.412 MOT Filter Mask 1 (B_CR_MOT_FILTER_MASK1_0_0_0_MCHBAR) — Offset 6BB8h

This register contains the value of the MOT 1 access filter Mask address.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6BB8h	7FFFFFFFE0 h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_0):** Reserved
38:5	3FFFFFFFFh RW	**MOT 1 Filter Mask Address (MOT_1_FILTER_MASK_ADDRESS):** Access address bits to mask. A value of 1 ignores the corresponding address bit. Half cache line granularity bits 38:5 to support legacy devices.

Bit Range	Default & Access	Field Name (ID): Description
4:0	0h RO	**Reserved (RESERVED_1):** Reserved

3.413 MOT Filter Misc 0 (B_CR_MOT_FILTER_MISC0_0_0_0_MCHBAR) — Offset 6BC0h

This register contains the value of the MOT 0 access filter Match/Mask Agent LPID.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6BC0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:35	0h RO	**Reserved (RESERVED_0):** Reserved
34:19	0h RW	**IA Core Match (IA_CORE_MATCH):** Match if transaction originated from one of up to 16 IA cores (one bit per logical processor).
18	0h RW	**GT Match (GT_MATCH):** • 1: Match • 0: Ignore
17:2	0h RW	**VC Match (VC_MATCH):** Match if transaction originated from one of up to 16 virtual channel cores (one bit per VC).
1	0h RW	**Filter Match Polarity (FILTER_MATCH_POLARITY):** • 1: Filter inversion • 0: Match criteria
0	0h RW	**Access Type -- Match Read or Write (ACCESS_TYPE_MATCH_READ_OR_WRITE):** • 1: Match writes • 0: Match reads

3.414 MOT Filter Misc 1 (B_CR_MOT_FILTER_MISC1_0_0_0_MCHBAR) — Offset 6BC8h

This register contains the value of the MOT 1 access filter Match/Mask Agent LPID.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6BC8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:35	0h RO	**Reserved (RESERVED_0):** Reserved
34:19	0h RW	**IA Core Match (IA_CORE_MATCH):** Match if transaction originated from one of up to 16 IA cores (one bit per logical processor).
18	0h RW	**GT Match (GT_MATCH):** • 1: Match • 0: Ignore
17:2	0h RW	**VC Match (VC_MATCH):** Match if transaction originated from one of up to 16 virtual channel cores (one bit per VC).
1	0h RW	**Filter Match Polarity (FILTER_MATCH_POLARITY):** • 1: Filter inversion • 0: Match criteria
0	0h RW	**Access Type -- Match Read or Write (ACCESS_TYPE_MATCH_READ_OR_WRITE):** • 1: Match writes • 0: Match reads

3.415 MOT Trigger Match 0 (B_CR_MOT_TRIGGER_MATCH0_0_0_0_MCHBAR) — Offset 6BD0h

This register contains the value of the MOT 0 access trigger Match address.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6BD0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_0):** Reserved
38:5	0h RW	**MOT 0 Trigger Match Address (MOT_0_TRIGGER_MATCH_ADDRESS):** Access address to match. Match results in trace of access. Half cache line granularity bits 38:5 to support legacy devices.
4:0	0h RO	**Reserved (RESERVED_1):** Reserved

3.416 MOT Trigger Mask 0 (B_CR_MOT_TRIGGER_MASK0_0_0_0_MCHBAR) — Offset 6BD8h

This register contains the value of the MOT 0 access trigger Mask address.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6BD8h	7FFFFFFFE0 h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_0):** Reserved
38:5	3FFFFFFFFh RW	**MOT 0 Trigger Mask Address (MOT_0_TRIGGER_MASK_ADDRESS):** Access address bits to mask. A value of 1 ignores the corresponding address bit. Half cache line granularity bits 38:5 to support legacy devices.
4:0	0h RO	**Reserved (RESERVED_1):** Reserved

3.417 MOT Trigger Match 1 (B_CR_MOT_TRIGGER_MATCH1_0_0_0_MCHBAR) — Offset 6BE0h

This register contains the value of the MOT 1 access trigger Match address.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6BE0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_0):** Reserved
38:5	0h RW	**MOT1 Trigger Match Address (MOT_1_TRIGGER_MATCH_ADDRESS):** Access address to match. Match results in trace of access. Half cache line granularity bits 38:5 to support legacy devices.
4:0	0h RO	**Reserved (RESERVED_1):** Reserved

3.418 MOT Trigger Mask 1 (B_CR_MOT_TRIGGER_MASK1_0_0_0_MCHBAR) — Offset 6BE8h

This register contains the value of the MOT 1 access trigger Mask address.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6BE8h	7FFFFFFFE0 h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_0):** Reserved
38:5	3FFFFFFFFh RW	**MOT 1 Trigger Mask Address (MOT_1_TRIGGER_MASK_ADDRESS):** Access address bits to mask. A value of 1 ignores the corresponding address bit. Half cache line granularity bits 38:5 to support legacy devices.
4:0	0h RO	**Reserved (RESERVED_1):** Reserved

3.419 MOT Trigger Misc 0 (B_CR_MOT_TRIGGER_MISC0_0_0_0_MCHBAR) — Offset 6BF0h

This register contains the value of the MOT 0 access trigger Match/Mask Agent LPID.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6BF0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:35	0h RO	**Reserved (RESERVED_0):** Reserved
34:19	0h RW	**IA Core Match (IA_CORE_MATCH):** Match if transaction originated from one of up to 16 IA cores (one bit per logical processor).
18	0h RW	**GT Match (GT_MATCH):** • 1: Match • 0: Ignore
17:2	0h RW	**VC Match (VC_MATCH):** Match if transaction originated from one of up to 16 virtual channel cores (one bit per VC).
1	0h RW	**Filter Match Polarity (FILTER_MATCH_POLARITY):** • 1: Filter inversion • 0: Match criteria
0	0h RW	**Access Type -- Match Read or Write (ACCESS_TYPE_MATCH_READ_OR_WRITE):** • 1: Match writes • 0: Match reads

3.420 MOT Trigger Misc 1 (B_CR_MOT_TRIGGER_MISC1_0_0_0_MCHBAR) — Offset 6BF8h

This register contains the value of the MOT 1 access trigger Match/Mask Agent LPID.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6BF8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:35	0h RO	**Reserved (RESERVED_0):** Reserved
34:19	0h RW	**IA Core Match (IA_CORE_MATCH):** Match if transaction originated from one of up to 16 IA cores (one bit per logical processor).
18	0h RW	**GT Match (GT_MATCH):** • 1: Match • 0: Ignore
17:2	0h RW	**VC Match (VC_MATCH):** Match if transaction originated from one of up to 16 virtual channel cores (one bit per VC).
1	0h RW	**Filter Match Polarity (FILTER_MATCH_POLARITY):** • 1: Filter inversion • 0: Match criteria
0	0h RW	**Access Type -- Match Read or Write (ACCESS_TYPE_MATCH_READ_OR_WRITE):** • 1: Match writes • 0: Match reads

3.421 MOT PSMI Sync (B_CR_MOT_PSMI_SYNC_0_0_0_MCHBAR) — Offset 6C00h

This register is used for PSMI sync.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6C00h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Reserved (RESERVED_0):** Reserved
1	0h RW/V	**PSMI Sync 1 (PSMI_SYNC1):** Set by PSMI microcode. Once set, cleared by HW after creation of first MOT record in Slice1.
0	0h RW/V	**PSMI Sync 0 (PSMI_SYNC0):** Set by PSMI microcode. Once set, cleared by HW after creation of first MOT record in Slice0.

3.422 B_CR_BIOSWR_CP (B_CR_BIOSWR_CP_0_0_0_MCHBAR) — Offset 6C08h

This register specifies the control policy for the BIOSWR security policy group. It controls write access to the Read Access Policy BIOSWR_RAC, Write Access Policy BIOSWR_WAC configuration registers, and self-referentially to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6C08h	40001000202 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_63):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
62	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_62):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
61	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_61):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
60	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_60):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
59	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_59):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
58	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_58):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
57	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_57):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
56	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_56):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
55	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_55):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
54	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_54):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
53	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_53):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
52	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_52):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
51	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_51):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
50	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_50):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
49	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_49):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
48	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_48):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
47	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_47):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
46	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_46):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
45	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_45):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
44	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_44):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
43	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_43):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
42	1h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_42):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
41	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_41):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
40	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_40):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
39	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_39):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
38	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_38):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
37	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_37):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
36	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_36):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
35	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_35):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
34	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_34):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
33	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_33):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
32	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_32):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
31	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_31):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
30	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_30):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
29	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_29):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
28	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_28):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
27	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_27):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
26	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_26):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
25	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_25):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
24	1h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_24):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
23	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_23):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
22	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_22):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
21	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_21):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
20	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_20):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
19	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_19):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
18	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_18):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
17	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_17):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
16	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_16):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
15	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_15):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
14	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_14):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
13	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_13):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
12	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_12):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
11	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_11):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
10	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_10):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
9	1h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_9):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
8	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_8):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
7	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_7):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
6	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_6):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
5	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_5):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
4	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_4):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
3	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_3):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
2	0h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_2):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
1	1h RW	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_1):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.
0	0h RO	**Memory Range Control Policy (MEM_RANGE_CTRL_POL_0):** Bit vector used to determine which agents are allowed write access to the BIOSWR_RAC, BIOSWR_WAC, and BIOSWR_CP registers, based on the value from each agent's 6bit SAI field.

3.423 BIOSWR Read Access Policy (B_CR_BIOSWR_RAC_0_0_0_MCHBAR) — Offset 6C10h

This register configures the Read Access Policy for the BIOSWR security policy group. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine read access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6C10h	80000C0063 0D0217 h

Bit Range	Default & Access	Field Name (ID): Description
63	1h RW	**Memory Range Policy (MEM_RANGE_POL_63):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
62	0h RW	**Memory Range Policy (MEM_RANGE_POL_62):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
61	0h RW	**Memory Range Policy (MEM_RANGE_POL_61):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
60	0h RW	**Memory Range Policy (MEM_RANGE_POL_60):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
59	0h RW	**Memory Range Policy (MEM_RANGE_POL_59):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
58	0h RW	**Memory Range Policy (MEM_RANGE_POL_58):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
57	0h RW	**Memory Range Policy (MEM_RANGE_POL_57):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
56	0h RW	**Memory Range Policy (MEM_RANGE_POL_56):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
55	0h RW	**Memory Range Policy (MEM_RANGE_POL_55):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
54	0h RW	**Memory Range Policy (MEM_RANGE_POL_54):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
53	0h RW	**Memory Range Policy (MEM_RANGE_POL_53):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
52	0h RW	**Memory Range Policy (MEM_RANGE_POL_52):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
51	0h RW	**Memory Range Policy (MEM_RANGE_POL_51):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
50	0h RW	**Memory Range Policy (MEM_RANGE_POL_50):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
49	0h RW	**Memory Range Policy (MEM_RANGE_POL_49):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
48	0h RW	**Memory Range Policy (MEM_RANGE_POL_48):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
47	0h RW	**Memory Range Policy (MEM_RANGE_POL_47):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
46	0h RW	**Memory Range Policy (MEM_RANGE_POL_46):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
45	0h RW	**Memory Range Policy (MEM_RANGE_POL_45):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
44	0h RW	**Memory Range Policy (MEM_RANGE_POL_44):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
43	1h RW	**Memory Range Policy (MEM_RANGE_POL_43):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
42	1h RW	**Memory Range Policy (MEM_RANGE_POL_42):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
41	0h RW	**Memory Range Policy (MEM_RANGE_POL_41):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
40	0h RW	**Memory Range Policy (MEM_RANGE_POL_40):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
39	0h RW	**Memory Range Policy (MEM_RANGE_POL_39):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
38	0h RW	**Memory Range Policy (MEM_RANGE_POL_38):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
37	0h RW	**Memory Range Policy (MEM_RANGE_POL_37):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
36	0h RW	**Memory Range Policy (MEM_RANGE_POL_36):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
35	0h RW	**Memory Range Policy (MEM_RANGE_POL_35):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
34	0h RW	**Memory Range Policy (MEM_RANGE_POL_34):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
33	0h RW	**Memory Range Policy (MEM_RANGE_POL_33):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
32	0h RW	**Memory Range Policy (MEM_RANGE_POL_32):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
31	0h RW	**Memory Range Policy (MEM_RANGE_POL_31):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
30	1h RW	**Memory Range Policy (MEM_RANGE_POL_30):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
29	1h RW	**Memory Range Policy (MEM_RANGE_POL_29):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
28	0h RW	**Memory Range Policy (MEM_RANGE_POL_28):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
27	0h RW	**Memory Range Policy (MEM_RANGE_POL_27):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
26	0h RW	**Memory Range Policy (MEM_RANGE_POL_26):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
25	1h RW	**Memory Range Policy (MEM_RANGE_POL_25):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
24	1h RW	**Memory Range Policy (MEM_RANGE_POL_24):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
23	0h RW	**Memory Range Policy (MEM_RANGE_POL_23):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
22	0h RW	**Memory Range Policy (MEM_RANGE_POL_22):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
21	0h RW	**Memory Range Policy (MEM_RANGE_POL_21):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
20	0h RW	**Memory Range Policy (MEM_RANGE_POL_20):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
19	1h RW	**Memory Range Policy (MEM_RANGE_POL_19):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
18	1h RW	**Memory Range Policy (MEM_RANGE_POL_18):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
17	0h RW	**Memory Range Policy (MEM_RANGE_POL_17):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
16	1h RW	**Memory Range Policy (MEM_RANGE_POL_16):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
15	0h RW	**Memory Range Policy (MEM_RANGE_POL_15):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
14	0h RW	**Memory Range Policy (MEM_RANGE_POL_14):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
13	0h RW	**Memory Range Policy (MEM_RANGE_POL_13):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**Memory Range Policy (MEM_RANGE_POL_12):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
11	0h RW	**Memory Range Policy (MEM_RANGE_POL_11):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
10	0h RW	**Memory Range Policy (MEM_RANGE_POL_10):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
9	1h RW	**Memory Range Policy (MEM_RANGE_POL_9):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
8	0h RW	**Memory Range Policy (MEM_RANGE_POL_8):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
7	0h RW	**Memory Range Policy (MEM_RANGE_POL_7):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
6	0h RW	**Memory Range Policy (MEM_RANGE_POL_6):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
5	0h RW	**Memory Range Policy (MEM_RANGE_POL_5):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
4	1h RW	**Memory Range Policy (MEM_RANGE_POL_4):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
3	0h RW	**Memory Range Policy (MEM_RANGE_POL_3):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
2	1h RW	**Memory Range Policy (MEM_RANGE_POL_2):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
1	1h RW	**Memory Range Policy (MEM_RANGE_POL_1):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.
0	1h RW	**Memory Range Policy (MEM_RANGE_POL_0):** Bit vector used to determine which agents are allowed read access, based on each agent's 6bit encoded SAI value.

3.424 B_CR_BIOSWR_WAC (B_CR_BIOSWR_WAC_0_0_0_MCHBAR) — Offset 6C18h

This register configures the Write Access Policy for the BIOSWR security policy group. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6C18h	400010C0212 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**Memory Range Policy (MEM_RANGE_POL_63):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
62	0h RW	**Memory Range Policy (MEM_RANGE_POL_62):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
61	0h RW	**Memory Range Policy (MEM_RANGE_POL_61):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
60	0h RW	**Memory Range Policy (MEM_RANGE_POL_60):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
59	0h RW	**Memory Range Policy (MEM_RANGE_POL_59):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
58	0h RW	**Memory Range Policy (MEM_RANGE_POL_58):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
57	0h RW	**Memory Range Policy (MEM_RANGE_POL_57):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
56	0h RW	**Memory Range Policy (MEM_RANGE_POL_56):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
55	0h RW	**Memory Range Policy (MEM_RANGE_POL_55):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
54	0h RW	**Memory Range Policy (MEM_RANGE_POL_54):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
53	0h RW	**Memory Range Policy (MEM_RANGE_POL_53):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
52	0h RW	**Memory Range Policy (MEM_RANGE_POL_52):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
51	0h RW	**Memory Range Policy (MEM_RANGE_POL_51):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
50	0h RW	**Memory Range Policy (MEM_RANGE_POL_50):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
49	0h RW	**Memory Range Policy (MEM_RANGE_POL_49):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
48	0h RW	**Memory Range Policy (MEM_RANGE_POL_48):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
47	0h RW	**Memory Range Policy (MEM_RANGE_POL_47):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
46	0h RW	**Memory Range Policy (MEM_RANGE_POL_46):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
45	0h RW	**Memory Range Policy (MEM_RANGE_POL_45):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
44	0h RW	**Memory Range Policy (MEM_RANGE_POL_44):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
43	0h RW	**Memory Range Policy (MEM_RANGE_POL_43):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
42	1h RW	**Memory Range Policy (MEM_RANGE_POL_42):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
41	0h RW	**Memory Range Policy (MEM_RANGE_POL_41):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
40	0h RW	**Memory Range Policy (MEM_RANGE_POL_40):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
39	0h RW	**Memory Range Policy (MEM_RANGE_POL_39):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
38	0h RW	**Memory Range Policy (MEM_RANGE_POL_38):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
37	0h RW	**Memory Range Policy (MEM_RANGE_POL_37):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
36	0h RW	**Memory Range Policy (MEM_RANGE_POL_36):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
35	0h RW	**Memory Range Policy (MEM_RANGE_POL_35):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
34	0h RW	**Memory Range Policy (MEM_RANGE_POL_34):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
33	0h RW	**Memory Range Policy (MEM_RANGE_POL_33):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
32	0h RW	**Memory Range Policy (MEM_RANGE_POL_32):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
31	0h RW	**Memory Range Policy (MEM_RANGE_POL_31):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
30	0h RW	**Memory Range Policy (MEM_RANGE_POL_30):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
29	0h RW	**Memory Range Policy (MEM_RANGE_POL_29):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
28	0h RW	**Memory Range Policy (MEM_RANGE_POL_28):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
27	0h RW	**Memory Range Policy (MEM_RANGE_POL_27):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
26	0h RW	**Memory Range Policy (MEM_RANGE_POL_26):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
25	0h RW	**Memory Range Policy (MEM_RANGE_POL_25):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
24	1h RW	**Memory Range Policy (MEM_RANGE_POL_24):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
23	0h RW	**Memory Range Policy (MEM_RANGE_POL_23):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
22	0h RW	**Memory Range Policy (MEM_RANGE_POL_22):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
21	0h RW	**Memory Range Policy (MEM_RANGE_POL_21):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
20	0h RW	**Memory Range Policy (MEM_RANGE_POL_20):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
19	1h RW	**Memory Range Policy (MEM_RANGE_POL_19):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
18	1h RW	**Memory Range Policy (MEM_RANGE_POL_18):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
17	0h RW	**Memory Range Policy (MEM_RANGE_POL_17):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
16	0h RW	**Memory Range Policy (MEM_RANGE_POL_16):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
15	0h RW	**Memory Range Policy (MEM_RANGE_POL_15):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
14	0h RW	**Memory Range Policy (MEM_RANGE_POL_14):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**Memory Range Policy (MEM_RANGE_POL_13):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
12	0h RW	**Memory Range Policy (MEM_RANGE_POL_12):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
11	0h RW	**Memory Range Policy (MEM_RANGE_POL_11):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
10	0h RW	**Memory Range Policy (MEM_RANGE_POL_10):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
9	1h RW	**Memory Range Policy (MEM_RANGE_POL_9):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
8	0h RW	**Memory Range Policy (MEM_RANGE_POL_8):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
7	0h RW	**Memory Range Policy (MEM_RANGE_POL_7):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
6	0h RW	**Memory Range Policy (MEM_RANGE_POL_6):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
5	0h RW	**Memory Range Policy (MEM_RANGE_POL_5):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
4	1h RW	**Memory Range Policy (MEM_RANGE_POL_4):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
3	0h RW	**Memory Range Policy (MEM_RANGE_POL_3):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
2	0h RW	**Memory Range Policy (MEM_RANGE_POL_2):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
1	1h RW	**Memory Range Policy (MEM_RANGE_POL_1):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.
0	0h RW	**Memory Range Policy (MEM_RANGE_POL_0):** Bit vector used to determine which agents are allowed write access, based on each agent's 6bit encoded SAI value.

3.425 Destination Override for BIOS1 region (B_CR_BIOS1_DECODE_OVERRIDE_0_0_0_MCHBAR) — Offset 6C20h

Specifies the controls to steer BIOS1 region to an alternate destination instead of the CSE when secure boot mode is enabled

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6C20h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RESERVED_0):** Reserved
3:0	0h RW	**OVERRIDE:** Bit 0: When set overrides CSE SRAM destination. Bits 3:1 reserved.

3.426 TPM Selector (B_CR_TPM_SELECTOR_0_0_0_MCHBAR) — Offset 6C24h

Specifies where the TPM lives in the platform. B-Unit uses this register to source decode for requests whose addresses fall within the fixed address range for the TPM.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6C24h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Reserved (RESERVED_0):** Reserved
1:0	0h RW	**TPM Selector (TPM_SELECTOR):** • 0h: fTPM enabled. Target of TPM accesses is the CSE. • 1h: SPI TPM enabled. Target of TPM accesses is SPI. • 2h: LPC TPM enabled. Target of TPM accesses is LPC. • 3h: All TPMs disabled. Target of TPM accesses is the PSF Error Handler.

3.427 B-Unit Pcode/Ucode Write, All Read Control Policy Register (B_CR_P_U_CODEWR_ALLRD_CP_0_0_0_MCHBAR) — Offset 6C28h

This register controls the access policy to the Read Access Policy P_U_CODEWR_ALLRD_RAC and Write Access Policy P_U_CODEWR_ALLRD_WAC configuration registers, and self-referentially to itself. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine both read access and write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6C28h	4000100020 2 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_63):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
62	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_62):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
61	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_61):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
60	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_60):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
59	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_59):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
58	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_58):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
57	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_57):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
56	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_56):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
55	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_55):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
54	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_54):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
53	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_53):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
52	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_52):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
51	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_51):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
50	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_50):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
49	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_49):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
48	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_48):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
47	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_47):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
46	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_46):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
45	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_45):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
44	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_44):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
43	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_43):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
42	1h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_42):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
41	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_41):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
40	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_40):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
39	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_39):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
38	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_38):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
37	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_37):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
36	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_36):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
35	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_35):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
34	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_34):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
33	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_33):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
32	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_32):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
31	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_31):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
30	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_30):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
29	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_29):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
28	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_28):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
27	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_27):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
26	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_26):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
25	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_25):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
24	1h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_24):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
23	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_23):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
22	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_22):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
21	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_21):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
20	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_20):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
19	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_19):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
18	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_18):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
17	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_17):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
16	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_16):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_15):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
14	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_14):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
13	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_13):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
12	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_12):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
11	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_11):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
10	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_10):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
9	1h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_9):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
8	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_8):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
7	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_7):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
6	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_6):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
5	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_5):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
4	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_4):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
3	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_3):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
2	0h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_2):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.
1	1h RW	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_1):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**B-Unit Pcode/Ucode Write, All Read Control Policy (IA_CTRL_POL_0):** Bit vector used to determine which agents are allowed access to the P_U_CODEWR_ALLRD_RAC, P_U_CODEWR_ALLRD_WAC, and P_U_CODEWR_ALLRD_CP registers, based on the value from the agent's 6bit SAI field.

3.428 B-Unit Pcode/Ucode Read Access Policy (B_CR_P_U_CODEWR_ALLRD_RAC_0_0_0_MCHBAR) — Offset 6C30h

This register configures the Read Access Policy for the B-Unit Pcode/Ucode Write, All Read registers.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6C30h	FFFFFFFFFFFFFFD7 h

Bit Range	Default & Access	Field Name (ID): Description
63	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_63):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
62	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_62):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
61	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_61):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
60	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_60):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
59	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_59):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
58	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_58):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
57	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_57):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
56	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_56):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
55	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_55):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
54	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_54):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
53	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_53):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
52	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_52):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
51	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_51):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
50	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_50):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
49	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_49):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
48	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_48):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
47	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_47):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
46	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_46):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
45	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_45):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
44	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_44):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
43	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_43):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
42	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_42):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
41	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_41):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
40	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_40):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
39	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_39):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
38	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_38):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
37	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_37):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
36	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_36):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
35	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_35):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
34	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_34):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
33	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_33):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
32	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_32):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
31	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_31):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
30	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_30):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
29	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_29):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
28	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_28):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
27	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_27):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
26	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_26):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
25	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_25):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
24	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_24):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
23	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_23):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
22	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_22):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
21	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_21):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
20	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_20):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
19	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_19):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
18	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_18):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
17	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_17):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
16	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_16):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
15	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_15):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
14	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_14):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
13	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_13):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
12	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_12):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
11	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_11):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
10	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_10):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
9	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_9):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
8	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_8):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
7	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_7):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
6	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_6):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
5	0h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_5):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
4	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_4):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
3	0h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_3):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
2	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_2):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
1	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_1):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.
0	1h RO	**B-Unit Pcode/Ucode Write, All Read SAI Read Access Policy (IA_SAI_POL_0):** Bit vector used to determine which agents are allowed read access to the B-Unit Pcode/Ucode Write, All Read policy registers, based on each agent's 6bit encoded SAI value.

3.429 B-Unit Pcode/Ucode Write Access Policy (B_CR_P_U_CODEWR_ALLRD_WAC_0_0_0_MCHBAR) — Offset 6C38h

This register configures the Write Access Policy for the B-Unit Pcode/Ucode Write, All Read policy registers. The requesting agent's 6bit encoded SAI value is used as an index into this register's bits to determine write access.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6C38h	4000100020 2 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_63):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
62	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_62):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
61	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_61):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
60	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_60):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
59	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_59):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
58	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_58):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
57	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_57):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
56	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_56):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
55	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_55):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
54	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_54):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
53	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_53):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
52	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_52):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
51	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_51):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
50	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_50):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
49	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_49):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
48	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_48):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
47	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_47):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
46	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_46):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
45	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_45):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
44	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_44):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
43	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_43):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
42	1h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_42):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
41	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_41):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
40	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_40):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
39	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_39):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
38	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_38):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
37	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_37):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
36	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_36):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
35	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_35):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
34	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_34):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
33	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_33):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
32	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_32):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
31	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_31):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
30	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_30):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
29	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_29):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
28	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_28):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
27	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_27):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
26	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_26):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
25	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_25):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
24	1h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_24):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
23	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_23):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
22	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_22):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
21	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_21):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
20	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_20):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
19	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_19):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_18):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
17	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_17):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
16	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_16):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
15	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_15):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
14	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_14):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
13	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_13):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
12	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_12):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
11	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_11):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
10	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_10):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
9	1h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_9):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
8	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_8):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
7	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_7):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
6	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_6):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
5	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_5):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
4	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_4):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_3):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
2	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_2):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
1	1h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_1):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.
0	0h RW	**B-Unit Pcode/Ucode Write, All Read SAI Write Access Policy (IA_SAI_POL_0):** Bit vector used to determine which agents are allowed write access to the B-Unit IA Core Configuration registers, based on each agent's 6bit encoded SAI value.

3.430 Default VTd BAR (B_CR_DEFVTDBAR_0_0_0_MCHBAR) — Offset 6C80h

BIOS must write DEFVTDBAR and then immediately follow it up with a read to DEFVTDBAR to ensure that all copies of DEFVTDBAR in the system are updated.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6C80h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:40	0h RO	**Reserved (RESERVED_0):** Reserved
39	0h RO	**DEFVTDBAR 40 Bit (DEFVTDBAR_40_BIT):** Reserved for future growth to 40bit addressing in uServer.
38:12	0h RW	**Default IOMMU VTd Config Space Base (DEFVTDBAR):** If DEFVTDBAR is enabled, this field corresponds to bits 38:12 of the base address default IOMMU VTd configuration space. BIOS will program this register resulting in a base address for a 4KB block of contiguous memory address space. This register ensures that a naturally aligned 4KB space is allocated within the first 512GB of addressable memory space. System Software uses this base address to program the default VTd IOMMU register set. If DEFVTDBAR is enabled and incoming Request Address[38:12] matches DEFVTDBAR[38:12] the request targets the Default VTd BAR.
11:2	0h RO	**Reserved (RESERVED_1):** Reserved
1	0h RW	**LOCK:** Locks the contents of the register including itself. Unused by the B-Unit, and does not implement the intended lock functionality. B-Unit includes this bit to support shadow copies of the register that rely on this lock bit.
0	0h RW/L	**DEFVTDBAR Enable (DEFVTDBAREN):** • 0: DEFVTDBAR is disabled and does not claim any memory • 1: DEFVTDBAR memory mapped accesses are claimed and decoded appropriately. This bit will remain 0 if VTd capability is disabled.

3.431 Gfx VTd BAR (B_CR_GFXVTDBAR_0_0_0_MCHBAR) — Offset 6C88h

BIOS must write GFXVTDBAR and then immediately follow it up with read of DEVEN_0_0_0_PCI, then write (with value from read operation) to DEVEN_0_0_0_PCI to ensure that all copies of GFXVTDBAR in the system are updated.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6C88h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_0):** Reserved
38:12	0h RW	**GFXVT Base Address (GFXVTBAR):** This field corresponds to bits 38 to 12 of the base address GFXVT configuration space. BIOS will program this register, resulting in a base address for a 4KB block of contiguous memory address space. This register ensures that a naturally aligned 4KB space is allocated within the first 512GB of addressable memory space. System Software uses this base address to program the GFXVT register set. If GFXVTBAR is enabled and incoming Request Address[38:12] matches GFXVTBAR[38:12] the request targets the Gfx VTd BAR.
11:2	0h RO	**Reserved (RESERVED_1):** Reserved
1	0h RW	**LOCK:** Locks the contents of the register, including itself. Unused by the B-Unit, and does not implement the intended lock functionality. B-Unit includes this bit to support shadow copies of the register that rely on this lock bit.
0	0h RW/L	**GFXVTBAR Enable (GFXVTBAREN):** • 0: GFXVTBAR is disabled and does not claim any memory. • 1: GFXVTBAR memory mapped accesses are claimed and decoded appropriately. This bit will remain 0 if VTd capability is disabled.

3.432 B-Unit Lites Group 0 Control (B_CR_LITES0_CTL_0_0_0_MCHBAR) — Offset 6C90h

This register controls the functionality of the B-Unit Lites Group 0 mask/match functionality.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6C90h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RW	**IDI Length Match (IDI_LENGTH_MATCH):** Enables length matching for IDI requests for Lites Group 0. Each bit, when set to 1, enables match for that length for an IDI request. Lites logic supports only lengths 0-16B for PRd and PortIn and only 64B for all other opcodes. • Bit 31:64B • Bit 30:16B • Bit 29:15B • Bit 28:14B • Bit 27:13B • Bit 26:12B • Bit 25:11B • Bit 24:10B • Bit 23:9B • Bit 22:8B • Bit 21:7B • Bit 20:6B • Bit 19:5B • Bit 18:4B • Bit 17:3B • Bit 16:2B • Bit 15:1B • Bit 14:0B
13:12	0h RW	**PII Length Match (PII_LENGTH_MATCH):** Enables length matching for PII requests for Lites Group 0. Each bit, when set to 1, enables length match for the PII request. • Bit 13:64B • Bit 12:32B
11:10	0h RO	**Reserved (RESERVED_0):** Reserved
9:8	0h RW	**Slice Match (SLICE_MATCH):** • 00: All mask/match hits are suppressed • 01: Mask/Match hits enabled only for Slice0 • 10: Mask/match hits enabled only for Slice1 • 11: Mask/Match hits enabled for both Slices
7:4	0h RW	**DWord Select (DWORD_SELECT):** Selects the dword within the 512bit data field that is compared for data mask/matches for Lites Group 0.
3	0h RW	**Enable Data Match (ENABLE_DATA_MATCH):** When set, this field enables data matching on Lites Group 0.
2	0h RW	**Alternate U2C Request View (ALTU2CREQVIEW):** When set, address match/ mask, opcode match and agent match registers are used to generate matches for the U2C request observation point. When clear, Badmit observation point matches are reported on the U2C request observation point.
1	0h RW	**Invert Address Match (INVERT_ADDR_MATCH):** When set, inverts the polarity of the address match/mask logic, i.e., reports a match for addresses that are not in the specified range.
0	0h RW	**Enable Group (ENABLE_GROUP):** Enables all match filters for Lites Group 0. The following are the observation points that contain match filters: Badmit logic within each slice after a transaction is successfully admitted into the B-Unit, U2C request launch, PMI datain for each PMI channel for read data, PMI dataout for each PMI channel for writes, data write from requesters to BRAM in each slice, and read data return on the live bypass and nonlive bypass paths in each slice. To report a match at an observation point, all match criteria for that point must be satisfied.

3.433 B-Unit Lites Group 0 Opcode Match Filter (B_CR_LITES0_OPCODE_MATCH_0_0_0_MCHBAR) — Offset 6C94h

This register contains the IDI and PII opcodes that will enable a Lites Group0 opcode match on a transaction.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6C94h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Opcode Match (OPCODE_MATCH):** Bit vector that enables a match on the opcode for C2U IDI requests, PII A2B requests, and U2C IDI requests. All three observation points, Badmit in both slices and the U2C request interface, use this same opcode match register. Each bit, when set, enables a match on the corresponding opcode, and, when clear, will suppress a match on the corresponding opcode. To match on any opcode, set all bits to 1. • Bit 31: C2U_Req_CRd • Bit 30: C2U_Req_DRd • Bit 29: C2U_Req_DRdPTE • Bit 28: C2U_Req_SetMonitor • Bit 27: C2U_Req_RFO,U2C_Req_LTWrite_PicletReplay • Bit 26: C2U_Req_PRd,U2C_Req_VLW_PicletReplay • Bit 25: C2U_Req_UcRdF • Bit 24: C2U_Req_PortIn,U2C_Req_SnpCode • Bit 23: C2U_Req_IntA,U2C_Req_SnpData • Bit 22: C2U_Req_Lock • Bit 21: C2U_Req_SplitLock,U2C_Req_IntLog_PicletReplay • Bit 20: C2U_Req_Unlock,U2C_Req_IntPhy_PicletReplay • Bit 19: C2U_Req_ItoM • Bit 18: C2U_Req_SpCyc,U2C_Req_SnpInv • Bit 17: C2U_Req_RdMonitor,U2C_Req_StopReq • Bit 16: C2U_Req_ClrMonitor,U2C_Req_StartReq • Bit 15: C2U_Req_CLFlush • Bit 14: C2U_Req_WbMtoI,U2C_Req_IntLog_MSI_noPicletReplay • Bit 13: C2U_Req_WbMtoE,U2C_Req_IntPhy_MSI_noPicletReplay • Bit 12: C2U_Req_WiL • Bit 11: C2U_Req_WCiL • Bit 10: C2U_Req_WCilF • Bit 9: C2U_Req_PortOut • Bit 8: C2U_Req_IntPriUp,U2C_Req_LTWrite_noPicletReplay • Bit 7: C2U_Req_IntLog • Bit 6: C2U_Req_IntPhy • Bit 5: C2U_Req_EOI,U2C_Req_VLW_noPicletReplay • Bit 4: C2U_Req_ItoMWr • Bit 3: A2B_Req_SnoopedRead • Bit 2: A2B_Req_UnSnoopedRead,U2C_Req_IntPhy_IPI_noPicletReplay • Bit 1: A2B_Req_SnoopedWrite • Bit 0: A2B_Req_UnSnoopedWrite,U2C_Req_IntLog_IPI_noPicletReplay

3.434 B-Unit Lites Group 0 Agent Match Filter (B_CR_LITES0_AGENT_MATCH_0_0_0_MCHBAR) — Offset 6C98h

This register designates which agents are enabled to trigger a Lites Group0 AgentID match on a transaction.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6C98h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GT Match (GT_MATCH):** This field is used to match IDI transactions from GT for Lites Group0. When set, enables a match for a transaction originating from GT. When clear, suppresses a match for a transaction from GT.
30:16	0h RW	**VC Match (VC_MATCH):** This field is used to match VC Channel ID for PII transactions for Lites Group0. Each bit, when set, enables a match for a transaction originating from that VC. When clear, suppresses a match for a transaction from that VC. • Bit 30: VC14 • Bit 29: VC13 • Bit 28: VC12 • Bit 27: VC11 • Bit 26: VC10 • Bit 25: VC9 • Bit 24: VC8 • Bit 23: VC7 • Bit 22: VC6 • Bit 21: VC5 • Bit 20: VC4 • Bit 19: VC3 • Bit 18: VC2 • Bit 17: VC1 • Bit 16: VC0
15:0	0h RW	**CPU Core Match (CPU_CORE_MATCH):** This field is used to match Logical Processor Core ID for CPU IDI transactions for Lites Group 0. Each bit, when set, enables a match for a transaction originating from that core. When clear, suppresses a match for a transaction from that core. • Bit 15: CPU7 Core1 • Bit 14: CPU7 Core0 • Bit 13: CPU6 Core1 • Bit 12: CPU6 Core0 • Bit 11: CPU5 Core1 • Bit 10: CPU5 Core0 • Bit 9: CPU4 Core1 • Bit 8: CPU4 Core0 • Bit 7: CPU3 Core1 • Bit 6: CPU3 Core0 • Bit 5: CPU2 Core1 • Bit 4: CPU2 Core0 • Bit 3: CPU1 Core1 • Bit 2: CPU1 Core0 • Bit 1: CPU0 Core1 • Bit 0: CPU0 Core0

3.435 B-Unit Lites Group 0 U2C IntData Match Filter (B_CR_LITES0_U2CINTDATA_MATCH_0_0_0_MCHBAR) — Offset 6C9Ch

When U2C alternate view is enabled, this register specifies match criteria for U2C IntData.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6C9Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Mask for U2C Request IntData[15:0] (INTDATA_MASK):** This mask is for generating Lites Group 0 U2C request alternative view match. If the mask bit in this register is 0, then the corresponding bit in the INTDATA_MATCH register is ignored. If the mask bit is 1, then the corresponding bit in the INTDATA_MATCH register must match the corresponding u2c request IntData bit for a match.
15:0	0h RW	**IntData Match (INTDATA_MATCH):** U2C Request IntData[15:0] value is compared with this field to generate a match for the U2C request address match.

3.436 B-Unit Lites Group 0 Address Match Filter LITES0_ADDR_MATCH (B_CR_LITES0_ADDR_MATCH_0_0_0_MCHBAR) — Offset 6CA0h

This register, together with the LITES0_ADDR_MASK register, specifies the request address values that trigger a Lites Group 0 address match on a transaction.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6CA0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_1):** Reserved
38:3	0h RW	**Address Match (ADDRESS_MATCH):** Address value to match for Lites.
2:0	0h RO	**Reserved (RESERVED_0):** Reserved

3.437 B-Unit Lites Group 0 Address Mask Filter LITES0_ADDR_MASK

(B_CR_LITES0_ADDR_MASK_0_0_0_MCHBAR) — Offset 6CA8h

This register, together with the LITES0_ADDR_MATCH register, specifies the request address values that trigger a Lites Group 0 address match on a transaction.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6CA8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_1):** Reserved
38:3	0h RW	**Address Mask (ADDRESS_MASK):** Address mask value used for comparing request address during filter operation for Lites. If the mask bit in this register is 0, then the corresponding bit in the ADDR_MATCH register is ignored. If the mask bit is 1, then the corresponding bit in the ADDR_MATCH register must match the corresponding request address bit for a match.
2:0	0h RO	**Reserved (RESERVED_0):** Reserved

3.438 B-Unit Lites Group 0 Data Match Filter LITES0_DATA_MATCH (B_CR_LITES0_DATA_MATCH_0_0_0_MCHBAR) — Offset 6CB0h

This register, together with LITES0_DATA_MASK, specifies the data values that will trigger a Lites Group 0 data filter match. All data observation points -- PMI data in, PMI data out, read data to agent (both live and nonlive) and write data from agent -- use the same Group 0 data match/mask registers for generating a match. Data match/mask can be enabled for only one DW of the transaction data. The specific DW can be selected via DWORD_SELECT.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6CB0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Data Match (DATA_MATCH):** Data value to match for Lites.

3.439 B-Unit Lites Group 0 Data Mask Filter LITES0_DATA_MASK (B_CR_LITES0_DATA_MASK_0_0_0_MCHBAR) — Offset 6CB4h

This register, together with LITES0_DATA_MATCH, specifies the data values that will trigger a Lites Group 0 data filter match for Lites. All data observation points -- PMI data in, PMI data out, read data to agent (both live and nonlive) and write data from agent -- use the same Group 0 data match/mask registers for generating a Group 0 data filter match. Data match/mask can be enabled for only one DW of the transaction data. The specific DW can be selected via DWORD_SELECT.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6CB4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Data Mask (DATA_MASK):** Data mask value used for comparing data during filter operations for Lites. If the mask bit in this register is 0, then the corresponding bit in the DATA_MATCH register is ignored. If the mask bit is 1, then the corresponding bit in the DATA_MATCH register must match the corresponding bit of the request data for a match.

3.440 B-Unit Lites Group 1 Control (B_CR_LITES1_CTL_0_0_0_MCHBAR) — Offset 6CC0h

This register controls the B-Unit Lites Group 1 mask/match functionality.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6CC0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RW	**IDI Length Match (IDI_LENGTH_MATCH):** Enables length matching for IDI requests for Lites Group 1. Each bit, when set to 1, enables match for that length for an IDI request. Lites logic supports only lengths 0-16B for PRd and PortIn and only 64B for all other opcodes. • Bit 31:64B • Bit 30:16B • Bit 29:15B • Bit 28:14B • Bit 27:13B • Bit 26:12B • Bit 25:11B • Bit 24:10B • Bit 23:9B • Bit 22:8B • Bit 21:7B • Bit 20:6B • Bit 19:5B • Bit 18:4B • Bit 17:3B • Bit 16:2B • Bit 15:1B • Bit 14:0B
13:12	0h RW	**PII Length Match (PII_LENGTH_MATCH):** Enables length matching for PII requests for Lites Group 1. Each bit, when set to 1, enables length match for the PII request. • Bit 13:64B • Bit 12:32B
11:10	0h RO	**Reserved (RESERVED_0):** Reserved
9:8	0h RW	**Slice Match (SLICE_MATCH):** • 00: All mask/match hits are suppressed • 01: mask/match hits enabled only for Slice0 • 10: mask/match hits enabled only for Slice1 • 11: mask/match hits enabled for both Slices
7:4	0h RW	**DWord Select (DWORD_SELECT):** Selects the dword within the 512bit data field that is compared for data mask/matches for Lites Group 1.
3	0h RW	**Enable Data Match (ENABLE_DATA_MATCH):** When set, this field enables data matching on Lites Group 1.
2	0h RW	**Alternate U2C Request View (ALTU2CREQVIEW):** When set, address match/mask opcode match and agent match registers are used to generate matches for the U2C request observation point. When clear, Badmit observation point matches are reported on the U2C request observation point.
1	0h RW	**Invert Address Match (INVERT_ADDR_MATCH):** When set, inverts the polarity of the address match/mask logic, i.e., reports a match for addresses that are not in the specified range.
0	0h RW	**Enable Group (ENABLE_GROUP):** Enables all match filters for Lites Group 1. The following are the observation points that contain match filters: Badmit logic within each slice after a transaction is successfully admitted into the B-Unit, U2C request launch, PMI datain for each PMI channel for read data, PMI dataout for each PMI channel for writes, data write from requesters to BRAM in each slice, and read data return on the live bypass and nonlive bypass paths in each slice. To report a match at an observation point, all match criteria for that point must be satisfied.

3.441 B-Unit Lites Group 1 Opcode Match Filter (B_CR_LITES1_OPCODE_MATCH_0_0_0_MCHBAR) — Offset 6CC4h

This register contains the IDI and PII opcodes that will enable a Lites Group1 opcode match on a transaction.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6CC4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Opcode Match (OPCODE_MATCH):** Bit vector that enables a match on the opcode for C2U IDI requests, PII A2B requests, and U2C IDI requests. All three observation points -- Badmit in both slices, and the U2C request interface -- use this same opcode match register. Each bit, when set, enables a match on the corresponding opcode, and when clear will suppress a match on the corresponding opcode. To match on any opcode, set all bits to 1. • Bit 31: C2U_Req_CRd • Bit 30: C2U_Req_DRd • Bit 29: C2U_Req_DRdPTE • Bit 28: C2U_Req_SetMonitor • Bit 27: C2U_Req_RFO,U2C_Req_LTWrite_PicletReplay • Bit 26: C2U_Req_PRd,U2C_Req_VLW_PicletReplay • Bit 25: C2U_Req_UcRdF • Bit 24: C2U_Req_PortIn,U2C_Req_SnpCode • Bit 23: C2U_Req_IntA,U2C_Req_SnpData • Bit 22: C2U_Req_Lock • Bit 21: C2U_Req_SplitLock,U2C_Req_IntLog_PicletReplay • Bit 20: C2U_Req_Unlock,U2C_Req_IntPhy_PicletReplay • Bit 19: C2U_Req_ItoM • Bit 18: C2U_Req_SpCyc,U2C_Req_SnpInv • Bit 17: C2U_Req_RdMonitor,U2C_Req_StopReq • Bit 16: C2U_Req_ClrMonitor,U2C_Req_StartReq • Bit 15: C2U_Req_CLFlush • Bit 14: C2U_Req_WbMtoI,U2C_Req_IntLog_MSI_noPicletReplay • Bit 13: C2U_Req_WbMtoE,U2C_Req_IntPhy_MSI_noPicletReplay • Bit 12: C2U_Req_WiL • Bit 11: C2U_Req_WCiL • Bit 10: C2U_Req_WCilF • Bit 9: C2U_Req_PortOut • Bit 8: C2U_Req_IntPriUp,U2C_Req_LTWrite_noPicletReplay • Bit 7: C2U_Req_IntLog • Bit 6: C2U_Req_IntPhy • Bit 5: C2U_Req_EOI,U2C_Req_VLW_noPicletReplay • Bit 4: C2U_Req_ItoMWr • Bit 3: A2B_Req_SnoopedRead • Bit 2: A2B_Req_UnSnoopedRead,U2C_Req_IntPhy_IPI_noPicletReplay • Bit 1: A2B_Req_SnoopedWrite • Bit 0: A2B_Req_UnSnoopedWrite,U2C_Req_IntLog_IPI_noPicletReplay

3.442 B-Unit Lites Group 1 Agent Match Filter (B_CR_LITES1_AGENT_MATCH_0_0_0_MCHBAR) — Offset 6CC8h

This register designates which agents are enabled to trigger a Lites Group1 AgentID match on a transaction.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6CC8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GT Match (GT_MATCH):** This field is used to match IDI transactions from GT for Lites Group1. When set, enables a match for a transaction originating from GT. When clear, suppresses a match for a transaction from GT.
30:16	0h RW	**VC Match (VC_MATCH):** This field is used to match VC Channel ID for PII transactions for Lites Group1. Each bit, when set, enables a match for a transaction originating from that VC. When clear, each bit suppresses a match for a transaction from that VC. • Bit 30: VC14 • Bit 29: VC13 • Bit 28: VC12 • Bit 27: VC11 • Bit 26: VC10 • Bit 25: VC9 • Bit 24: VC8 • Bit 23: VC7 • Bit 22: VC6 • Bit 21: VC5 • Bit 20: VC4 • Bit 19: VC3 • Bit 18: VC2 • Bit 17: VC1 • Bit 16: VC0
15:0	0h RW	**CPU Core Match (CPU_CORE_MATCH):** This field is used to match Logical Processor Core ID for CPU IDI transactions, for Lites Group 1. Each bit, when set, enables a match for a transaction originating from that core. When clear, each bit suppresses a match for a transaction from that core. • Bit 15: CPU7 Core1 • Bit 14: CPU7 Core0 • Bit 13: CPU6 Core1 • Bit 12: CPU6 Core0 • Bit 11: CPU5 Core1 • Bit 10: CPU5 Core0 • Bit 9: CPU4 Core1 • Bit 8: CPU4 Core0 • Bit 7: CPU3 Core1 • Bit 6: CPU3 Core0 • Bit 5: CPU2 Core1 • Bit 4: CPU2 Core0 • Bit 3: CPU1 Core1 • Bit 2: CPU1 Core0 • Bit 1: CPU0 Core1 • Bit 0: CPU0 Core0

3.443 B-Unit Lites Group 1 U2C IntData Match Filter (B_CR_LITES1_U2CINTDATA_MATCH_0_0_0_MCHBAR) — Offset 6CCCh

When U2C alternate view is enabled, this register specifies match criteria for U2C IntData.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6CCCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**IntData Mask (INTDATA_MASK):** Mask for U2C Request IntData[15:0] for generating Lites Group 1 U2C request alternative view match. If the mask bit in this register is 0, then the corresponding bit in the INTDATA_MATCH register is ignored. If the mask bit is 1, then the corresponding bit in the INTDATA_MATCH register must match the corresponding u2c request IntData bit for a match.
15:0	0h RW	**IntData Match (INTDATA_MATCH):** U2C Request IntData[15:0] value is compared with this field to generate a match for the U2C request address match.

3.444 B-Unit Lites Group 1 Address Match Filter LITES1_ADDR_MATCH (B_CR_LITES1_ADDR_MATCH_0_0_0_MCHBAR) — Offset 6CD0h

This register, together with the LITES1_ADDR_MASK register, specifies the request address values that trigger a Lites Group 1 address match on a transaction.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6CD0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_1):** Reserved
38:3	0h RW	**Address Match (ADDRESS_MATCH):** Address value to match for Lites
2:0	0h RO	**Reserved (RESERVED_0):** Reserved

3.445 B-Unit Lites Group 1 Address Mask Filter LITES1_ADDR_MASK

(B_CR_LITES1_ADDR_MASK_0_0_0_MCHBAR) — Offset 6CD8h

This register, together with the LITES1_ADDR_MATCH register, specifies the request address values that trigger a Lites Group 1 address match on a transaction.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6CD8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_1):** Reserved
38:3	0h RW	**Address Mask (ADDRESS_MASK):** Address mask value used for comparing request address during filter operation for Lites. If the mask bit in this register is 0, then the corresponding bit in the ADDR_MATCH register is ignored. If the mask bit is 1, then the corresponding bit in the ADDR_MATCH register must match the corresponding request address bit for a match.
2:0	0h RO	**Reserved (RESERVED_0):** Reserved

3.446 B-Unit Lites Group 1 Data Match Filter LITES1_DATA_MATCH (B_CR_LITES1_DATA_MATCH_0_0_0_MCHBAR) — Offset 6CE0h

This register, together with LITES1_DATA_MASK, specifies the data values that will trigger a Lites Group 1 data filter match. All data observation points -- PMI data in, PMI data out, read data to agent (both live and nonlive) and write data from agent -- use the same Group 1 data match/mask registers for generating a match. Data match/mask can be enabled for only one DW of the transaction data. The specific DW can be selected via DWORD_SELECT.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6CE0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Data Match (DATA_MATCH):** Data value to match for Lites.

3.447 B-Unit Lites Group 1 Data Mask Filter LITES1_DATA_MASK (B_CR_LITES1_DATA_MASK_0_0_0_MCHBAR) — Offset 6CE4h

This register, together with LITES1_DATA_MATCH, specifies the data values that will trigger a Lites Group 1 data filter match for Lites. All data observation points -- PMI data in, PMI data out, read data to agent (both live and nonlive) and write data from agent -- use the same Group 1 data match/mask registers for generating a Group 1 data filter match. Data match/mask can be enabled for only one DW of the transaction data. The specific DW can be selected via DWORD_SELECT.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6CE4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Data Mask (DATA_MASK):** Data mask value used for comparing data during filter operations for Lites. If the mask bit in this register is 0, then the corresponding bit in the DATA_MATCH register is ignored. If the mask bit is 1, then the corresponding bit in the DATA_MATCH register must match the corresponding bit of the request data for a match.

3.448 B-Unit Lites Group 2 Control (B_CR_LITES2_CTL_0_0_0_MCHBAR) — Offset 6CF0h

This register controls the functionality of the B-Unit Lites Group 2 mask/match functionality.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6CF0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RW	**IDI Length Match (IDI_LENGTH_MATCH):** Enables length matching for IDI requests for Lites Group 2. Each bit, when set to 1, enables match for that length for an IDI request. Lites logic supports only lengths 0-16B for PRd and PortIn and only 64B for all other opcodes. • Bit 31:64B • Bit 30:16B • Bit 29:15B • Bit 28:14B • Bit 27:13B • Bit 26:12B • Bit 25:11B • Bit 24:10B • Bit 23:9B • Bit 22:8B • Bit 21:7B • Bit 20:6B • Bit 19:5B • Bit 18:4B • Bit 17:3B • Bit 16:2B • Bit 15:1B • Bit 14:0B
13:12	0h RW	**PII Length Match (PII_LENGTH_MATCH):** Enables length matching for PII requests for Lites Group 2. Each bit, when set to 1, enables length match for the PII request. • Bit 13:64B • Bit 12:32B
11:10	0h RO	**Reserved (RESERVED_0):** Reserved
9:8	0h RW	**Slice Match (SLICE_MATCH):** • 00: All mask/match hits are suppressed • 01: Mask/Match hits enabled only for Slice0 • 10: Mask/match hits enabled only for Slice1 • 11: Mask/Match hits enabled for both Slices
7:4	0h RW	**Dword Select (DWORD_SELECT):** Selects the dword within the 512bit data field that is compared for data mask/matches for Lites Group 2.
3	0h RW	**Enable Data Match (ENABLE_DATA_MATCH):** When set, this field enables data matching on Lites Group 2.
2	0h RW	**Alternname UC2 Request View (ALTU2CREQVIEW):** When set, address match/ mask opcode match and agent match registers are used to generate matches for the U2C request observation point. When clear, Badmit observation point matches are reported on the U2C request observation point.
1	0h RW	**Invert Address Match (INVERT_ADDR_MATCH):** When set, inverts the polarity of the address match/mask logic, i.e., reports a match for addresses that are not in the specified range.
0	0h RW	**Enable Group (ENABLE_GROUP):** Enables all match filters for Lites Group 2. The following are the observation points that contain match filters: Badmit logic within each slice after a transaction is successfully admitted into the B-Unit, U2C request launch PMI datain for each PMI channel read data, PMI dataout for each PMI channel write data, agent data write to BRAM in each slice, read data return on the live bypass and nonlive bypass paths in each slice. To report a match at an observation point all match criteria for that point must be satisfied.

3.449 B-Unit Lites Group 2 Opcode Match Filter (B_CR_LITES2_OPCODE_MATCH_0_0_0_MCHBAR) — Offset 6CF4h

This register contains the IDI and PII opcodes that will enable a Lites Group2 opcode match on a transaction.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6CF4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Opcode Match (OPCODE_MATCH):** Bit vector that enables a match on the opcode for C2U IDI requests, PII A2B requests, and U2C IDI requests. All three observation points -- Badmit in both slices, and the U2C request interface -- use this same opcode match register. Each bit, when set, enables a match on the corresponding opcode and when clear will suppress a match on the corresponding opcode. To match on any opcode, set all bits to 1. • Bit 31: C2U_Req_CRd • Bit 30: C2U_Req_DRd • Bit 29: C2U_Req_DRdPTE • Bit 28: C2U_Req_SetMonitor • Bit 27: C2U_Req_RFO,U2C_Req_LTWrite_PicletReplay • Bit 26: C2U_Req_PRd,U2C_Req_VLW_PicletReplay • Bit 25: C2U_Req_UcRdF • Bit 24: C2U_Req_PortIn,U2C_Req_SnpCode • Bit 23: C2U_Req_IntA,U2C_Req_SnpData • Bit 22: C2U_Req_Lock • Bit 21: C2U_Req_SplitLock,U2C_Req_IntLog_PicletReplay • Bit 20: C2U_Req_Unlock,U2C_Req_IntPhy_PicletReplay • Bit 19: C2U_Req_ItoM • Bit 18: C2U_Req_SpCyc,U2C_Req_SnpInv • Bit 17: C2U_Req_RdMonitor,U2C_Req_StopReq • Bit 16: C2U_Req_ClrMonitor,U2C_Req_StartReq • Bit 15: C2U_Req_CLFlush • Bit 14: C2U_Req_WbMtoI,U2C_Req_IntLog_MSI_noPicletReplay • Bit 13: C2U_Req_WbMtoE,U2C_Req_IntPhy_MSI_noPicletReplay • Bit 12: C2U_Req_WiL • Bit 11: C2U_Req_WCiL • Bit 10: C2U_Req_WCilF • Bit 9: C2U_Req_PortOut • Bit 8: C2U_Req_IntPriUp,U2C_Req_LTWrite_noPicletReplay • Bit 7: C2U_Req_IntLog • Bit 6: C2U_Req_IntPhy • Bit 5: C2U_Req_EOI,U2C_Req_VLW_noPicletReplay • Bit 4: C2U_Req_ItoMWr • Bit 3: A2B_Req_SnoopedRead • Bit 2: A2B_Req_UnSnoopedRead,U2C_Req_IntPhy_IPI_noPicletReplay • Bit 1: A2B_Req_SnoopedWrite • Bit 0: A2B_Req_UnSnoopedWrite,U2C_Req_IntLog_IPI_noPicletReplay

3.450 B-Unit Lites Group 2 Agent Match Filter (B_CR_LITES2_AGENT_MATCH_0_0_0_MCHBAR) — Offset 6CF8h

This register designates which agents are enabled to trigger a Lites Group2 AgentID match on a transaction.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6CF8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GT Match (GT_MATCH):** This field is used to match IDI ransactions from GT for Lites Group2. When set, enables a match for a transaction originating from GT. When clear, suppresses a match for a transaction from GT.
30:16	0h RW	**VC Match (VC_MATCH):** This field is used to match VC Channel ID, for PII transactions for Lites Group2. Each bit, when set, enables a match for a transaction originating from that VC. When clear, suppresses a match for a transaction from that VC. • Bit 30: VC14 • Bit 29: VC13 • Bit 28: VC12 • Bit 27: VC11 • Bit 26: VC10 • Bit 25: VC9 • Bit 24: VC8 • Bit 23: VC7 • Bit 22: VC6 • Bit 21: VC5 • Bit 20: VC4 • Bit 19: VC3 • Bit 18: VC2 • Bit 17: VC1 • Bit 16: VC0
15:0	0h RW	**CPU Core Match (CPU_CORE_MATCH):** This field is used to match Logical Processor Core ID for CPU IDI transactions for Lites Group 2. Each bit, when set, enables a match for a transaction originating from that core. When clear, suppresses a match for a transaction from that core. • Bit 15: CPU7 Core1 • Bit 14: CPU7 Core0 • Bit 13: CPU6 Core1 • Bit 12: CPU6 Core0 • Bit 11: CPU5 Core1 • Bit 10: CPU5 Core0 • Bit 9: CPU4 Core1 • Bit 8: CPU4 Core0 • Bit 7: CPU3 Core1 • Bit 6: CPU3 Core0 • Bit 5: CPU2 Core1 • Bit 4: CPU2 Core0 • Bit 3: CPU1 Core1 • Bit 2: CPU1 Core0 • Bit 1: CPU0 Core1 • Bit 0: CPU0 Core0

3.451 B-Unit Lites Group 2 U2C IntData Match Filter (B_CR_LITES2_U2CINTDATA_MATCH_0_0_0_MCHBAR) — Offset 6CFCh

When U2C alternate view is enabled, this register specifies match criteria for U2C IntData.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6CFCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**IntData Mask (INTDATA_MASK):** Mask for U2C Request IntData[15:0] for generating Lites Group 2 U2C request alternative view match. If the mask bit in this register is 0, then the corresponding bit in the INTDATA_MATCH register is ignored. If the mask bit is 1, then the corresponding bit in the INTDATA_MATCH register must match the corresponding u2c request IntData bit for a match.
15:0	0h RW	**IntData Match (INTDATA_MATCH):** U2C Request IntData[15:0] value is compared with this field to generate a match for the U2C request address match.

3.452 B-Unit Lites Group 2 Address Match Filter LITES2_ADDR_MATCH (B_CR_LITES2_ADDR_MATCH_0_0_0_MCHBAR) — Offset 6D00h

This register, together with the LITES2_ADDR_MASK register, specifies the request address values that trigger a Lites Group 2 address match on a transaction.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6D00h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_1):** Reserved
38:3	0h RW	**Address Match (ADDRESS_MATCH):** Address value to match for Lites.
2:0	0h RO	**Reserved (RESERVED_0):** Reserved

3.453 B-Unit Lites Group 2 Address Mask Filter LITES2_ADDR_MASK

(B_CR_LITES2_ADDR_MASK_0_0_0_MCHBAR) — Offset 6D08h

This register, together with the LITES2_ADDR_MATCH register, specifies the request address values that trigger a Lites Group 2 address match on a transaction.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6D08h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_1):** Reserved
38:3	0h RW	**Address Mask (ADDRESS_MASK):** Address mask value used for comparing request address during filter operation for Lites. If the mask bit in this register is 0, then the corresponding bit in the ADDR_MATCH register is ignored. If the mask bit is 1, then the corresponding bit in the ADDR_MATCH register must match the corresponding request address bit for a match.
2:0	0h RO	**Reserved (RESERVED_0):** Reserved

3.454 B-Unit Lites Group 2 Data Match Filter LITES2_DATA_MATCH (B_CR_LITES2_DATA_MATCH_0_0_0_MCHBAR) — Offset 6D10h

This register, together with LITES2_DATA_MASK, specifies the data values that will trigger a Lites Group 2 data filter match. All data observation points PMI data in PMI data out read data to agent both live and nonlive and write data from agent use the same Group 2 data match/mask registers for generating a match. Data match/mask can be enabled for only one DW of the transaction data. The specific DW can be selected via DWORD_SELECT.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D10h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Data Match (DATA_MATCH):** Data value to match for Lites

3.455 B-Unit Lites Group 2 Data Mask Filter LITES2_DATA_MASK (B_CR_LITES2_DATA_MASK_0_0_0_MCHBAR) — Offset 6D14h

This register, together with LITES2_DATA_MATCH, specifies the data values that will trigger a Lites Group 2 data filter match for Lites. All data observation points PMI data in PMI data out read data to agent both live and nonlive and write data from agent use the same Group 2 data match/mask registers for generating a Group 2 data filter match. Data match/mask can be enabled for only one DW of the transaction data. The specific DW can be selected via DWORD_SELECT.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Data Mask (DATA_MASK):** Data mask value used for comparing data during filter operations for Lites. If the mask bit in this register is 0, then the corresponding bit in the DATA_MATCH register is ignored. If the mask bit is 1, then the corresponding bit in the DATA_MATCH register must match the corresponding bit of the request data for a match.

3.456 B-Unit Lites Group 3 Control (B_CR_LITES3_CTL_0_0_0_MCHBAR) — Offset 6D20h

This register controls the functionality of the B-Unit Lites Group 3 mask/match functionality.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D20h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RW	**IDI Length Match (IDI_LENGTH_MATCH):** Enables length matching for IDI requests for Lites Group 3. Each bit, when set to 1, enables match for that length for an IDI request. Lites logic supports only lengths 0-16B for PRd and PortIn and only 64B for all other opcodes. • Bit 31:64B • Bit 30:16B • Bit 29:15B • Bit 28:14B • Bit 27:13B • Bit 26:12B • Bit 25:11B • Bit 24:10B • Bit 23:9B • Bit 22:8B • Bit 21:7B • Bit 20:6B • Bit 19:5B • Bit 18:4B • Bit 17:3B • Bit 16:2B • Bit 15:1B • Bit 14:0B
13:12	0h RW	**PII Length Match (PII_LENGTH_MATCH):** Enables length matching for PII requests for Lites Group 3. Each bit, when set to 1, enables length match for the PII request: • Bit 13:64B • Bit 12:32B
11:10	0h RO	**Reserved (RESERVED_0):** Reserved
9:8	0h RW	**Slice Match (SLICE_MATCH):** • 00: All mask/match hits are suppressed • 01: Mask/match hits enabled only for Slice0 • 10: Mask/match hits enabled only for Slice1 • 11: Mask/match hits enabled for both Slices
7:4	0h RW	**Dword Select (DWORD_SELECT):** Selects the dword within the 512bit data field that is compared for data mask/matches for Lites Group 3.
3	0h RW	**Enable Data Match (ENABLE_DATA_MATCH):** When set, this field enables data matching on Lites Group 3.
2	0h RW	**Alternate U2C Request View (ALTU2CREQVIEW):** When set, address match/ mask, opcode match and agent match registers are used to generate matches for the U2C request observation point. When clear, Badmit observation point matches are reported on the U2C request observation point.
1	0h RW	**Invert Address Match (INVERT_ADDR_MATCH):** When set inverts the polarity of the address match/mask logic i.e. reports a match for addresses that are not in the specified range.
0	0h RW	**Enable Group (ENABLE_GROUP):** Enables all match filters for Lites Group 3. The following are the observation points that contain match filters Badmit logic within each slice after a transaction is successfully admitted into the B-Unit U2C request launch PMI datain for each PMI channel read data PMI dataout for each PMI channel write data agent data write to BRAM in each slice read data return on the live bypass and nonlive bypass paths in each slice. To report a match at an observation point all match criteria for that point must be satisfied.

3.457 B-Unit Lites Group 3 Opcode Match Filter (B_CR_LITES3_OPCODE_MATCH_0_0_0_MCHBAR) — Offset 6D24h

This register contains the IDI and PII opcodes that will enable a Lites Group3 opcode match on a transaction.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D24h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Opcode Match (OPCODE_MATCH):** Bit vector that enables a match on the opcode for C2U IDI requests, PII A2B requests, and U2C IDI requests. All three observation points -- Badmit in both slices, and the U2C request interface -- use this same opcode match register. Each bit, when set, enables a match on the corresponding opcode, and when clear will suppress a match on the corresponding opcode. To match on any opcode, set all bits to 1. • Bit 31: C2U_Req_CRd • Bit 30: C2U_Req_DRd • Bit 29: C2U_Req_DRdPTE • Bit 28: C2U_Req_SetMonitor • Bit 27: C2U_Req_RFO,U2C_Req_LTWrite_PicletReplay • Bit 26: C2U_Req_PRd,U2C_Req_VLW_PicletReplay • Bit 25: C2U_Req_UcRdF • Bit 24: C2U_Req_PortIn,U2C_Req_SnpCode • Bit 23: C2U_Req_IntA,U2C_Req_SnpData • Bit 22: C2U_Req_Lock • Bit 21: C2U_Req_SplitLock,U2C_Req_IntLog_PicletReplay • Bit 20: C2U_Req_Unlock,U2C_Req_IntPhy_PicletReplay • Bit 19: C2U_Req_ItoM • Bit 18: C2U_Req_SpCyc,U2C_Req_SnpInv • Bit 17: C2U_Req_RdMonitor,U2C_Req_StopReq • Bit 16: C2U_Req_ClrMonitor,U2C_Req_StartReq • Bit 15: C2U_Req_CLFlush • Bit 14: C2U_Req_WbMtoI,U2C_Req_IntLog_MSI_noPicletReplay • Bit 13: C2U_Req_WbMtoE,U2C_Req_IntPhy_MSI_noPicletReplay • Bit 12: C2U_Req_WiL • Bit 11: C2U_Req_WCiL • Bit 10: C2U_Req_WCilF • Bit 9: C2U_Req_PortOut • Bit 8: C2U_Req_IntPriUp,U2C_Req_LTWrite_noPicletReplay • Bit 7: C2U_Req_IntLog • Bit 6: C2U_Req_IntPhy • Bit 5: C2U_Req_EOI,U2C_Req_VLW_noPicletReplay • Bit 4: C2U_Req_ItoMWr • Bit 3: A2B_Req_SnoopedRead • Bit 2: A2B_Req_UnSnoopedRead,U2C_Req_IntPhy_IPI_noPicletReplay • Bit 1: A2B_Req_SnoopedWrite • Bit 0: A2B_Req_UnSnoopedWrite,U2C_Req_IntLog_IPI_noPicletReplay

3.458 B-Unit Lites Group 3 Agent Match Filter (B_CR_LITES3_AGENT_MATCH_0_0_0_MCHBAR) — Offset 6D28h

This register designates which agents are enabled to trigger a Lites Group3 AgentID match on a transaction.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D28h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GT Match (GT_MATCH):** This field is used to match IDI ransactions from GT for Lites Group3. When set, enables a match for a transaction originating from GT. When clear, suppresses a match for a transaction from GT.
30:16	0h RW	**VC Match (VC_MATCH):** This field is used to match VC Channel ID for PII transactions, for Lites Group3. Each bit, when set, enables a match for a transaction originating from that VC. When clear, the bit suppresses a match for a transaction from that VC. • Bit 30: VC14 • Bit 29: VC13 • Bit 28: VC12 • Bit 27: VC11 • Bit 26: VC10 • Bit 25: VC9 • Bit 24: VC8 • Bit 23: VC7 • Bit 22: VC6 • Bit 21: VC5 • Bit 20: VC4 • Bit 19: VC3 • Bit 18: VC2 • Bit 17: VC1 • Bit 16: VC0
15:0	0h RW	**CPU Core Match (CPU_CORE_MATCH):** This field is used to match Logical Processor Core ID for CPU IDI transactions, for Lites Group 3. Each bit, when set, enables a match for a transaction originating from that core. When clear, the bit suppresses a match for a transaction from that core. • Bit 15: CPU7 Core1 • Bit 14: CPU7 Core0 • Bit 13: CPU6 Core1 • Bit 12: CPU6 Core0 • Bit 11: CPU5 Core1 • Bit 10: CPU5 Core0 • Bit 9: CPU4 Core1 • Bit 8: CPU4 Core0 • Bit 7: CPU3 Core1 • Bit 6: CPU3 Core0 • Bit 5: CPU2 Core1 • Bit 4: CPU2 Core0 • Bit 3: CPU1 Core1 • Bit 2: CPU1 Core0 • Bit 1: CPU0 Core1 • Bit 0: CPU0 Core0

3.459 B-Unit Lites Group 3 U2C IntData Match Filter (B_CR_LITES3_U2CINTDATA_MATCH_0_0_0_MCHBAR) — Offset 6D2Ch

When U2C alternate view is enabled, this register specifies match criteria for U2C IntData.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**IntData Mask (INTDATA_MASK):** Mask for U2C Request IntData[15:0] for generating Lites Group 3 U2C request alternative view match. If the mask bit in this register is 0, then the corresponding bit in the INTDATA_MATCH register is ignored. If the mask bit is 1, then the corresponding bit in the INTDATA_MATCH register must match the corresponding u2c request IntData bit for a match.
15:0	0h RW	**IntData Match (INTDATA_MATCH):** U2C Request IntData[15:0] value is compared with this field to generate a match for the U2C request address match.

3.460 B-Unit Lites Group 3 Address Match Filter LITES3_ADDR_MATCH (B_CR_LITES3_ADDR_MATCH_0_0_0_MCHBAR) — Offset 6D30h

This register, together with the LITES3_ADDR_MASK register, specifies the request address values that trigger a Lites Group 3 address match on a transaction.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6D30h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_1):** Reserved
38:3	0h RW	**Address Match (ADDRESS_MATCH):** Address value to match for Lites.
2:0	0h RO	**Reserved (RESERVED_0):** Reserved

3.461 B-Unit Lites Group 3 Address Mask Filter LITES3_ADDR_MASK

(B_CR_LITES3_ADDR_MASK_0_0_0_MCHBAR) — Offset 6D38h

This register, together with the LITES3_ADDR_MATCH register, specifies the request address values that trigger a Lites Group 3 address match on a transaction.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 6D38h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_1):** Reserved
38:3	0h RW	**Address Mask (ADDRESS_MASK):** Address mask value used for comparing request address during filter operation for Lites. If the mask bit in this register is 0, then the corresponding bit in the ADDR_MATCH register is ignored. If the mask bit is 1, then the corresponding bit in the ADDR_MATCH register must match the corresponding request address bit for a match.
2:0	0h RO	**Reserved (RESERVED_0):** Reserved

3.462 B-Unit Lites Group 3 Data Match Filter LITES3_DATA_MATCH (B_CR_LITES3_DATA_MATCH_0_0_0_MCHBAR) — Offset 6D40h

This register, together with LITES3_DATA_MASK, specifies the data values that will trigger a Lites Group 3 data filter match. All data observation points -- PMI data in, PMI data out, read data to agent (both live and nonlive) and write data from agent -- use the same Group 3 data match/mask registers for generating a match. Data match/mask can be enabled for only one DW of the transaction data. The specific DW can be selected via DWORD_SELECT.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D40h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DATA_MATCH:** Data value to match for Lites

3.463 B-Unit Lites Group 3 Data Mask Filter LITES3_DATA_MASK (B_CR_LITES3_DATA_MASK_0_0_0_MCHBAR) — Offset 6D44h

This register, together with LITES3_DATA_MATCH, specifies the data values that will trigger a Lites Group 3 data filter match for Lites. All data observation points -- PMI data in, PMI data out, read data to agent (both live and nonlive) and write data from agent -- use the same Group 3 data match/mask registers for generating a Group 3 data filter match. Data match/mask can be enabled for only one DW of the transaction data. The specific DW can be selected via DWORD_SELECT.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D44h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Data Mask (DATA_MASK):** Data mask value used for comparing data during filter operations for Lites. If the mask bit in this register is 0, then the corresponding bit in the DATA_MATCH register is ignored. If the mask bit is 1, then the corresponding bit in the DATA_MATCH register must match the corresponding bit of the request data for a match.

3.464 B-Unit Lites and Emon Master Control LITESEMONCTL (B_CR_LITESEMON_CTL_0_0_0_MCHBAR) — Offset 6D48h

This register controls the functionality of the B-Unit Lites Debug functionality and Emon exposure to VISA.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D48h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Enable Lites (ENABLE_LITES):** • 0: Lites logic is disabled. All Lites Views will be driven to 0s. • 1: Lites logic is enabled.
30:27	0h RW	**Emon Barb CQ Select (EMON_BARB_CQ_SELECT):** Select barb conflictQ FIFO. • 0000 to 0111: pii0 to pii7 • 1xx0: idi Slice0 • 1xx1: idi Slice1

Bit Range	Default & Access	Field Name (ID): Description
26:24	0h RW	**Emon Barb Ingress Select (EMON_BARB_INGRESS_SELECT):** Select barb ingress FIFO. Bits 2:1 correspond to IDI attach, while bit 0 is for slice in the case that the IDI attach point has a per-slice FIFO. • 000: IDI attach point 0, Slice 0 (or mono_fifo if not using per-slice FIFO for this attach point) • 001: IDI attach point 0, Slice 1 • 010: IDI attach point 1, Slice 0 (or mono_fifo if not using per-slice FIFO for this attach point) • 011: IDI attach point 1, Slice 1 • 100: IDI attach point 2, Slice 0 (or mono_fifo if not using per-slice FIFO for this attach point) • 101: IDI attach point 2, Slice 1 • 110: IDI attach point 3, Slice 0 (or mono_fifo if not using per-slice FIFO for this attach point) • 111: IDI attach point 3, Slice 1
23:13	0h RW	**Emon VC Mask (EMON_VC_MASK):** B-Unit and T-Unit expose only the Emons corresponding to the VC(s) specified in this field (it is a mask).
12:5	0h RW	**Emon IDI Mask (EMON_IDI_MASK):** B-Unit and T-Unit expose only the Emons corresponding to the IDI agent(s) specified in this field (it is a mask). IDI agent refers to GLM modules and GT not Sunits.
4:2	0h RW	**Lites IDI Select (LITES_IDI_SELECT):** B-Unit and T-Unit expose only the views corresponding to the IDI agent specified in this field. IDI agents only refer to GT and GLM modules, not Sunits. Used only on the C2U response views.
1	0h RW	**Lites Slice Select (LITES_SLICE_SELECT):** When set to 0, B-Unit and T-Unit expose only Slice0 views. No Lites observability of any transaction routed to Slice 1. When set to 1, B-Unit and T-Unit expose only Slice1 views. No Lites observability of any transaction routed to Slice0.
0	0h RW	**Lites PMI Select (LITES_PMI_SELECT):** When set to 0, B-Unit exposes only PMI channel 0 views in either slice. No Lites observability of any transaction routed to PMI channel 1 in either slice. When set to 1, B-Unit exposes only PMI Channel 1 views in either slice.

3.465 B-Unit Arbiter Control BARBCTRL0 (B_CR_BARBCTRL0) — Offset 6D4Ch

Specifies the weighting for Agents 03 used by the B-Unit's Badmit Arbiter. The value specified in the Agent Weight field is used by the Badmit arbiters weight counters to determine the number of requests from an agent that are allowed to be granted before updating the requester's age register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D4Ch	4040404 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RESERVED_0):** Reserved
29:24	4h RW	**Agent 3 Weight (AGENT3_WEIGHT):** Arbiter weight for Agent 3.
23:22	0h RO	**Reserved (RESERVED_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
21:16	4h RW	**Agent 2 Weight (AGENT2_WEIGHT):** Arbiter weight for Agent 2.
15:14	0h RO	**Reserved (RESERVED_2):** Reserved
13:8	4h RW	**Agent 1 Weight (AGENT1_WEIGHT):** Arbiter weight for Agent 1.
7:6	0h RO	**Reserved (RESERVED_3):** Reserved
5:0	4h RW	**Agent 0 Weight (AGENT0_WEIGHT):** Arbiter weight for Agent 0.

3.466 B-Unit Arbiter Control BARBCTRL1 (B_CR_BARBCTRL1) — Offset 6D50h

Specifies the weighting for Agents 4-7 used by the B-Unit's Badmit Arbiter. The value specified in the Agent Weight field is used by the Badmit arbiter's weight counters to determine the number of requests from an agent that are allowed to be granted before updating the requester's age register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D50h	4040404 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RESERVED_0):** Reserved
29:24	4h RW	**Agent 7 Weight (AGENT7_WEIGHT):** Arbiter weight for Agent 7.
23:22	0h RO	**Reserved (RESERVED_1):** Reserved
21:16	4h RW	**Agent 6 Weight (AGENT6_WEIGHT):** Arbiter weight for Agent 6.
15:14	0h RO	**Reserved (RESERVED_2):** Reserved
13:8	4h RW	**Agent 5 Weight (AGENT5_WEIGHT):** Arbiter weight for Agent 5.
7:6	0h RO	**Reserved (RESERVED_3):** Reserved
5:0	4h RW	**Agent 4 Weight (AGENT4_WEIGHT):** Arbiter weight for Agent 4.

3.467 B-Unit Scheduler Control (B_CR_BSCHWT0) — Offset 6D54h

Specifies the weighting for Agents 0-3 used by the B-Unit Scheduler. The value is used by the B-Unit Scheduler to determine how many requests can be granted for the agent before the agent's requests are masked by the Scheduler Arbiter, to allow other agents' requests to be granted.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D54h	4040404 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RESERVED_0):** Reserved
29:24	4h RW	**Agent 3 Weight (AGENT3_WEIGHT):** Arbiter weight for Agent 3.
23:22	0h RO	**Reserved (RESERVED_1):** Reserved
21:16	4h RW	**Agent 2 Weight (AGENT2_WEIGHT):** Arbiter weight for Agent 2.
15:14	0h RO	**Reserved (RESERVED_2):** Reserved
13:8	4h RW	**Agent 1 Weight (AGENT1_WEIGHT):** Arbiter weight for Agent 1.
7:6	0h RO	**Reserved (RESERVED_3):** Reserved
5:0	4h RW	**Agent 0 Weight (AGENT0_WEIGHT):** Arbiter weight for Agent 0.

3.468 B-Unit Scheduler Control (B_CR_BSCHWT1) — Offset 6D58h

Specifies the weighting for Agents 4-7 used by the B-Unit Scheduler. The value is used by the B-Unit Scheduler to determine how many requests can be granted for the agent before the agent's requests are masked by the Scheduler Arbiter to allow other agents' requests to be granted.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D58h	4040404 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RESERVED_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
29:24	4h RW	**Agent 7 Weight (AGENT7_WEIGHT):** Arbiter weight for Agent 7.
23:22	0h RO	**Reserved (RESERVED_1):** Reserved
21:16	4h RW	**Agent 6 Weight (AGENT6_WEIGHT):** Arbiter weight for Agent 6.
15:14	0h RO	**Reserved (RESERVED_2):** Reserved
13:8	4h RW	**Agent 5 Weight (AGENT5_WEIGHT):** Arbiter weight for Agent 5.
7:6	0h RO	**Reserved (RESERVED_3):** Reserved
5:0	4h RW	**Agent 4 Weight (AGENT4_WEIGHT):** Arbiter weight for Agent 4.

3.469 B-Unit Scheduler Control (B_CR_BSCHWT2) — Offset 6D5Ch

Specifies the weighting for Agents 8-11 used by the B-Unit Scheduler. The value is used by the B-Unit Scheduler to determine how many requests can be granted for the agent before the agent's requests are masked by the Scheduler Arbiter to allow other agents' requests to be granted.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D5Ch	4040404 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RESERVED_0):** Reserved
29:24	4h RW	**Agent 11 Weight (AGENT11_WEIGHT):** Arbiter weight for Agent 11.
23:22	0h RO	**Reserved (RESERVED_1):** Reserved
21:16	4h RW	**Agent 10 Weight (AGENT10_WEIGHT):** Arbiter weight for Agent 10.
15:14	0h RO	**Reserved (RESERVED_2):** Reserved
13:8	4h RW	**Agent 9 Weight (AGENT9_WEIGHT):** Arbiter weight for Agent 9.
7:6	0h RO	**Reserved (RESERVED_3):** Reserved
5:0	4h RW	**Agent 8 Weight (AGENT8_WEIGHT):** Arbiter weight for Agent 8.

3.470 B-Unit Scheduler Control (B_CR_BSCHWT3) — Offset 6D60h

Specifies the weighting for Agents 12-15 used by the B-Unit Scheduler. The value is used by the B-Unit Scheduler to determine how many requests can be granted for the agent before the agent's requests are masked by the Scheduler Arbiter to allow other agents' requests to be granted.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D60h	4040404 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RESERVED_0):** Reserved
29:24	4h RW	**Agent 15 Weight (AGENT15_WEIGHT):** Arbiter weight for Agent 15.
23:22	0h RO	**Reserved (RESERVED_1):** Reserved
21:16	4h RW	**Agent 14 Weight (AGENT14_WEIGHT):** Arbiter weight for Agent 14.
15:14	0h RO	**Reserved (RESERVED_2):** Reserved
13:8	4h RW	**Agent 13 Weight (AGENT13_WEIGHT):** Arbiter weight for Agent 13.
7:6	0h RO	**Reserved (RESERVED_3):** Reserved
5:0	4h RW	**Agent 12 Weight (AGENT12_WEIGHT):** Arbiter weight for Agent 12.

3.471 B-Unit Flush Control (B_CR_BWFLUSH) — Offset 6D64h

Controls the policy used to determine when dirty entries must be flushed to DRAM. When the number of dirty entries is lower than a high watermark dirty_hwm the B-Unit will opportunistically flush data to DRAM after the write flush timeout value. When the number of dirty entries exceeds the high water mark, the B-Unit will initiate a highpriority flush and push dirty data to DRAM until the count is once again below the low water mark.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D64h	FF010000 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	FFh RW	**Flush Threshold (FLUSH_THRESHOLD):** All write commands are blocked at Badmit if the number of write commands in the Flush Pool exceeeds this value.

Bit Range	Default & Access	Field Name (ID): Description
23:17	0h RO	**Reserved (RESERVED_0):** Reserved
16	1h RO/V	**All Entries Flushed (ALL_ENTRIES_FLUSHED):** All dirty entries in the B-Unit, in both slices, have been flushed.
15:8	0h RW	**Dirty Low Water Mark (DIRTY_LWM):** Low water mark for dirty entries retained by the B-Unit. B-Unit will immediately attempt to flush any dirty entry, hence setting the low water mark to 0.
7:0	0h RW	**Dirty High Water Mark (DIRTY_HWM):** High water mark for dirty entries retained by the B-Unit. B-Unit will immediately attempt to flush any dirty entry, hence setting the low water mark to 0.

3.472 B-Unit Flush Weights (B_CR_BFLWT) — Offset 6D68h

Controls B-Unit alternate scheduling of reads and writes to DRAM.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D68h	400 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Disable Flush Weights (DISABLE_FLUSH_WEIGHTS):** When set to 1, disables flush weights. Flushing of dirty entries will start when Dirty Limit is HWM, and continue until Dirty Limit is LWM. No reads will be scheduled in between.
30:14	0h RO	**Reserved (RESERVED_1):** Reserved
13:8	4h RW	**Write Weights (WRITE_WEIGHTS):** Number of write requests sent to a PMI channel before switching to scheduling ISOC or non-ISOC read requests, when use of flush weights is not disabled. If the flush weights are enabled, this has to be set at least to 1 otherwise Bunit will not schedule any writes and hangs may occur.
7:0	0h RO	**Reserved (RESERVED_0):** Reserved

3.473 Weighted Scheduling Control of High Priority ISOC and Other Requests (B_CR_BISOCWT) — Offset 6D6Ch

Controls alternate scheduling of High Priority ISOC requests and other requests.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D6Ch	80003F0F h

Bit Range	Default & Access	Field Name (ID): Description
31	1h RW	**Enable ISOC Requests (ENABLE_ISOC_WEIGHTS):** When set to 1, enables switching from scheduling High Priority ISOC requests to scheduling Best Effort and Low Priority ISOC requests, based on ISOC weights and NONISOC weights.

Bit Range	Default & Access	Field Name (ID): Description
30:14	0h RO	**Reserved (RESERVED_1):** Reserved
13:8	3Fh RW	**ISOC Request Weights (ISOC_REQUEST_WEIGHTS):** Number of high priority isochronous requests sent to a PMI channel before switching to scheduling non-ISOC read or write requests, when use of ISOC weights is not disabled. If the ISOC weights are enabled, this has to be set to at least 1 otherwise Bunit will not schedule any ISOC reads and hangs may occur.
7:6	0h RO	**Reserved (RESERVED_0):** Reserved
5:0	Fh RW	**Non ISOC Request Weights (NON_ISOC_REQUEST_WEIGHTS):** Number of non-high priority isochronous and best effort requests sent to a PMI channel before switching to scheduling ISOC read or write requests, when use of ISOC weights is not disabled. If the ISOC weights are enabled, this has to be set to at least 1 otherwise Bunit will not schedule any non-ISOC reads and hangs may occur.

3.474 B-Unit Control (B_CR_BCTRL2) — Offset 6D70h

Contains basic control information used by the B-Unit.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D70h	F0034 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Enable Read Invalidate Timer (ENABLE_READ_INVALIDATE_TIMER):** When set to 1, enables the BRAM to clear the RD_DONE bit and flush dirty data when a timer expires. Used when parity is not enabled, to force the B-Unit to not indefinitely cache previously read lines, and also to cause a flush of dirty data that has been written to the BRAM entry.
30	0h RW	**Enable DRAM Read for Partial Write (ENABLE_DRAM_READ):** The B-Unit has byte enable capability, so it doesn't require a DRAM read when there is a partial write. This bit forces it to perform the read anyway.
29:26	0h RO	**Reserved (RESERVED_2):** Reserved
25	0h RW	**DRAM_ECC_ENABLE:** If DRAM ECC is enabled, the data error bits from the Dunit is used for indicating the derror for the corresponding transactions to the agents and forcing the poison bit when BUNIT PARITY is unsupported and Read for partial writes are enabled
24	0h RW	**MOT Disable Stall Arbiter on Error (MOT_DISABLE_STALL_ARB_ON_ERR):** If the HH widget gets an error, allow transactions to get arbitrated.
23:16	Fh RW	**BRAM Read Invalidate Time (BRAM_READ_INVALIDATE_TIME):** Timer threshold to clear rd_done bits or flush a BRAM entry, if dirty. The value specified is in multiples of 250ns. For each time interval specified, the read done status of one BRAM entry will be cleared based on a BTAG index which is then incremented to point to the next BRAM entry.
15:8	0h RW	**Casual Timer (CASUAL_TIMER):** The number of clock cycles that the B-Unit waits before starting a casual dirty flush. Casual Flush feature will not be enabled by PND2 architecture. Instead, a dirty write will be made eliGBle for scheduling immediately.
7:6	0h RO	**Reserved (RESERVED_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5	1h RW	**Enable 64B Write (ENABLE_64B_WRITE):** When set, B-unit will send 64B PMI Write requests for transactions requiring write access to DRAM. Must always be set to one, otherwise functional errors may occur.
4	1h RW	**Enable 64B Read (ENABLE_64B_READ):** When set, B-unit will send 64B PMI read requests for transactions requiring read access to DRAM. Must always be set to one, otherwise functional errors may occur.
3	0h RO	**Demand Scrub Enable (DEMAND_SCRUB_ENABLE):** This mode causes B-Unit to automatically issue a flush to PMI, thus writing correct data back to memory for any read request that returns with a correctable data error.
2	1h RW	**Enable Read Done for Write (ENABLE_READ_DONE_FOR_WRITE):** Enable Any writes (IWB or normal writes) to set the read_done bit in bstat, which enables return of data from the BRAM cache instead of Memory.
1	0h RW	**Miss Valid Entries (MISS_VALID_ENTRIES):** This mode causes reads to clean valid B-Unit buffer entries to look like misses instead of hits. When this bit is set -- even when all requested bytes are valid and present in the BRAM data buffer -- B-Unit will send read requests over the PMI interface to re-fetch the data from DRAM, instead of returning it from the BRAM.
0	0h RW	**Dirty Stall (DIRTY_STALL):** This mode causes reads or writes from any requester interface to dirty valid B-Unit buffer entries to stall on the appropriate requester interface until the entry has been flushed from the B-Unit.

3.475 Asset Classification Bits (B_CR_AC_RS0_0_0_0_MCHBAR) — Offset 6D74h

B-Unit Asset Classification AC[3] bits for IMRs and Special Protected Memory Region. Controls whether RS0 Root Space 0 transactions from PII are allowed access to IMRs and Protected Memory Region.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D74h	100000 h

Bit Range	Default & Access	Field Name (ID): Description
31:21	0h RO	**Reserved (RESERVED_0):** Reserved
20	1h RW	**MOT RS Asset Classification (MOT_RS0_EN):** RS Asset Classification bit for the MOT buffer. PII transactions from RS0 that hit the MOT buffer will be allowed access only when both of the following conditions are met: a) Request SAI is in the legal permitted list, as specified in the RAC/WAC policy registers, and b) MOT_RS0_EN bit is set to 1. PII RS0 transactions targeting DRAM that do not hit any enabled IMR or special protected regions will always be allowed access.
19:0	0h RW	**IMR RS Asset Classification (IMR_RS0_EN):** RS Asset Classification bit for IMRs 0-19. PII transactions from RS0 that hit an enabled IMR address range will be allowed access only when both of the following conditions are met: a) Request SAI is in the legal permitted list as specified in the IMRs RAC/WAC policy registers and b) IMR_AC_RS bit corresponding to the IMR is set to 1. PII RS0 transactions targeting DRAM that do not hit any enabled IMR or special protected regions will always be allowed access.

3.476 IDI Real-Time Feature Configuration Bits (B_CR_RT_EN_0_0_0_MCHBAR) — Offset 6D78h

IDI Real-Time Feature Configuration register. Controls which IDI attachpoint support Real-Time traffic/transactions.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D78h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RSVD):** Reserved
16	0h RW	**IDI Agent Real-Time Traffic Mask Bits (RT_IDI_AGENT):** IDI Agent Real-Time Traffic Mask Bits. Controls which IDI Agent supports Real-Time Traffic/Transactions. Only one IDI Agent needs to be selected for real-time traffic. Currently we don't support multiple IDI agents to be enabled for real-time traffic at the same time. This field will be active only when Bit 0:RT_ENABLE is set to '1'
15:1	0h RO	**Reserved (RSVD):** Reserved
0	0h RW	**Real-Time Enable (RT_ENABLE):** Global enable bit for Real-Time Support

3.477 B-Unit Control Register 3 (B_CR_BCTRL3) — Offset 6D7Ch

Specifies miscellaneous controls for the the B-Unit.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6D7Ch	40 h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Reserved (RSVD):** Reserved
8	0h RW	**Disable C2U Ingress Slice Anti-Starvation (SLICE_ANTI_STARVE_DISABLE):** When set, B-unit will always make an available request in each instantiated per-slice C2U Ingress FIFO for all IDI attach points eliGBle for arbitration in their respective odd/even sa2xclk clock. When clear, B-Unit will mask a C2U Ingress Slices available request from arbitration when SLICE_ANTI_STARVE_THRESHOLD consecutive requests has been granted for that IDI attach point from the other Slice Ingress FIFO. Has no effect for IDI attach points that do not have per-slice Ingress FIFOs.
7:0	40h RW	**C2U Ingress Slice Anti-Starvation Threshold (SLICE_ANTI_STARVE_THRESHOLD):** Specifies the threshold for the number of consecutive requests B-admit arbiter will grant from the same Slice ingress FIFO for an IDI attachpoint, while the other Slice also has a request available

3.478 Asymmetric Memory Region 0 (B_CR_ASYM_MEM_REGION0_0_0_0_MCHBAR) — Offset 6E40h

Specification of asymmetric memory region 0 (in slice 0) for the configuration with 2 asymmetric memory regions.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6E40h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Slice 0 Asymmetric Enable (SLICE0_ASYM_ENABLE):** Setting this bit to 0 disables asymmetric memory region 0; setting it to 1 enables the region.
30	0h RW	**Channel Select for ASYM Region Slice 0 (SLICE0_ASYM_CHANNEL_SELECT):** Specifies Channel Selected for ASYM Region Mapped To Slice 0
29:19	0h RW	**Slice 0 Asymmetric Limit (SLICE0_ASYM_LIMIT):** Specifies bits [38:28] of the highest address of asymmetric memory region 0 (in slice 0); all the lower bits of the region's highest address are equal to 1.
18:15	0h RO	**Reserved (RSVD):** Reserved
14:4	0h RW	**Slice 0 Asymmetric Base (SLICE0_ASYM_BASE):** Specifies bits [38:28] of the base address of asymmetric memory region 0 (in slice 0); all the lower bits of the region's base address are equal to 0.
3:0	0h RO	**Reserved (RSVD):** Reserved

3.479 Asymmetric Memory Region 1 (B_CR_ASYM_MEM_REGION1_0_0_0_MCHBAR) — Offset 6E44h

Specification of asymmetric memory region 1 (in slice 1) for the configuration with 2 asymmetric memory regions.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6E44h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Slice 1 Asymmetric Enable (SLICE1_ASYM_ENABLE):** Setting this bit to 0 disables asymmetric memory region 1; setting it to 1 enables the region.
30	0h RW	**Channel Select for ASYM SLICE 1 (SLICE1_ASYM_CHANNEL_SELECT):** Specifies Channel Selected for ASYM Region Mapped To Slice 1.
29:19	0h RW	**Slice 1 Asymmetric Limit (SLICE1_ASYM_LIMIT):** Specifies bits [38:28] of the highest address of asymmetric memory region 1 (in slice 1); all the lower bits of the region's highest address are equal to 1.

Bit Range	Default & Access	Field Name (ID): Description
18:15	0h RO	**Reserved (RSVD):** Reserved
14:4	0h RW	**Slice 1 Asymmetric Base (SLICE1_ASYM_BASE):** Specifies bits [38:28] of the base address of asymmetric memory region 1 (in slice 1); all the lower bits of the region's base address are equal to 0.
3:0	0h RO	**Reserved (RSVD):** Reserved

3.480 B-Unit Machine Check Mode Low (B_CR_BMCMODE_LOW) — Offset 6E48h

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6E48h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	**Reserved (RESERVED_0):** Reserved
0	0h RW	**Machine Check Signal Mode (MC_SIGNAL_MODE):** When set to 1, B-Unit will not allow any transaction with uncorrectable error or subsequent memory transaction to propagate through to Requester. This will essentially hang CPU and CPU will end up with IERR shutdown. Issue: When set to zero B-Unit will allow transaction with an uncorrectable error to propagate and signal MC event to CPU if enabled in IA32_MC5_CTL. If enabled MC event will be taken by CPU cores at the end of instruction boundary after it detected by ROB

3.481 B-Unit Machine Check Mode High (B_CR_BMCMODE_HIGH) — Offset 6E4Ch

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6E4Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RESERVED_0):** Reserved

3.482 Two-Way Asymmetric Memory Region Configuration (B_CR_ASYM_2WAY_MEM_REGION_0_0_0_MCHBAR) — Offset 6E50h

Specification of asymmetric memory region for the configuration with 2way Interleaved Asymmetric memory. It is only supported if all Slices and all Channels are Enabled

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6E50h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Enable Two-Way Asymmetric Memory Configuration (ASYM_2WAY_INTERLEAVE_ENABLE):** Setting this bit to 0 disables 2Way Asymmetric Interleaving; setting it to 1 enables the region.
30:28	0h RO	**Reserved (RSVD):** Reserved
27:17	0h RW	**Limit Address for Two-Way Asymmetric Memory Region (ASYM_2WAY_LIMIT):** Specifies bits [38:28] of the highest address of Interleave Asymmetric Region; all the lower bits of the region's highest address are equal to 1.
16:15	0h RO	**Reserved (RSVD):** Reserved
14:4	0h RW	**Base Address for Two-Way Asymmetric Memory Region (ASYM_2WAY_BASE):** Specifies bits [38:28] of the base address of Interleave Asymmetric Region; all the lower bits of the region's base address are equal to 0.
3:2	0h RW	**Two-Way Asymmetric Interleave Mode (ASYM_2WAY_INTLV_MODE):** Going with 2 bits here. 2'b00 : Asymmetric memory Split between Channel 0 of Slice 0 and Slice 1 2'b01 : Asymmetric memory split between Channel 1 of Slice 0 and Slice 1 2'b10 : Asymmetric memory split between Channel 0 and Channel 1 of Slice 0 2'b11 : Asymmetric memory split between Channel 0 and Channel 1 of Slice 1
1:0	0h RO	**Reserved (RSVD):** Reserved

3.483 SGX Status (B_CR_SGX_STATUS_0_0_0_MCHBAR) — Offset 6E54h

Status register that is writable by BUnit to give updates regarding various SGX events

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 6E54h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	**Reserved (RESERVED_0):** Reserved
0	0h RO/V	**PRMRR Flushed (PRMRR_FLUSHED):** This bit is used to indicate that all transactions that could target the PRMRR region before it was enabled have been flushed to memory. After enabling PRMRR, MCHECK will pole this bit and wait until it is high. 0b: Transactions remain in the BRAM that could target the PRMRR region, but were allocated before PRMRR SAI checks were enforced. 1b: All transactions in the BRAM have passed through the PRMRR SAI checks.

3.484 Thermal Device Mailbox Data0 (P_CR_THERMAL_MAILBOX_DATA0_0_0_0_MCHBAR) — Offset 7000h

This register represents the lower 32b of the thermal mailbox data. THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7000h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**DATA:** This field contains the low 32 bits of data associated with specific commands.

3.485 Thermal Device Mailbox Data1 (P_CR_THERMAL_MAILBOX_DATA1_0_0_0_MCHBAR) — Offset 7004h

This register represents the upper 32b of the thermal mailbox data. THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7004h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**DATA:** This field contains the low 32 bits of data associated with specific commands.

3.486 Thermal Device Mailbox Interface (P_CR_THERMAL_MAILBOX_INTERFACE_0_0_0_MCHBAR) — Offset 7008h

This register implements the control and response of the Thermal Device Mailbox. Software may use this mailbox to configure and query various parameters into SOC thermal / power control. This particular register is responsible for initiating requests to the thermal device and reading responses. THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7008h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S/V	**Run/Busy (RUN_BUSY):** The run/busy control is used for managing the semaphore on the mailbox interface. Typical usage involves the following flow: • Software waits for the interface run/busy bit to clear. • Software writes the mailbox data registers as appropriate for this command. • Software writes a command encoding and sets the run/busy bit in the interface register. • Software waits for the interface run/busy bit to clear to indicate the command has been handled. • Software queries the completion code in the command field of the interface register to ensure it passed (0b indicates pass) Bit encoding for the run/busy: • 0 = The thermal mailbox is idle or the last request has been completed. Software may initiate new requests. • 1 = The thermal mailbox is busy. The thermal device is still handling a request. Writes to thermal mailbox are not allowed at this time.
30:29	0h RO	**Reserved (RSVD):** Reserved
28:8	0h RW/V	**Additional Parameters (ADDR_CNTL):** This field is used as an additional modifier to the command encoding for incoming mailbox requests. In thermal device mailbox responses, this field is always zero. The applicability of this additional parameter field is handled on a case by case basis for the services supplied by this mailbox
7:0	0h RW/V	**COMMAND:** For incoming requests (where run/busy=1), this field represents the command opcode. For responses (where run/busy=0), this field represents the response completion code. A completion code of 0b indicates passing, all other completion codes indicate failure. For detailed definition of services provided by this mailbox, see the full mailbox specification.

3.487 Thermal Device IRQ and Lock Configuration (P_CR_THERMAL_DEVICE_IRQ_0_0_0_MCHBAR) — Offset 700Ch

IRQ vector number for thermal/power device that is sent to the IOAPIC and Lock field for INTR_LAT_0_0_1_PCI.INTRPIN

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 700Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/L	**LOCK:** Used to lock P_CR_INTR_LAT_0_0_1_PCI.INTPIN. BIOS should set this lock bit before passing control to the OS
30:8	0h RO	**RESERVED:** Reserved
7:0	0h RW	**IRQ:** IRQ vector number for the thermal / power device. This field controls the event vector issues to the IOAPIC. It must be configured by BIOS for INTA support. Recommended setting for Broxton is 24 (18h)

3.488 Package Thermal Interrupt Control (P_CR_PKG_THERM_INTERRUPT_0_0_0_MCHBAR) — Offset 7010h

This register is used to manage processor thermal interrupts, including management of filtering on the virtual thermal sensor control signal. These features are designed to allow software to implement smooth control of thermally significant events for platform thermal management. THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7010h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved
30:24	0h RW	**Time Window (TIME_WINDOW):** Virtual Temperature thermal filter RC time constant. Virtual temperature readings are run through an RC filter before they are fed into status and interrupt generation. This filtering allows software smooth control of thermal responses to thermally significant events. The bits of this field describe parameters for a mathematical equation for time window configuration. This field is split into two sub-fields: • x = bits[6:5] • y = bits[4:0] Time window equation: **time_window = PACKAGE_POWER_SKU_UNIT.TIME_UNIT * ((1+x/4)^y)**
23:16	0h RW	**Thermal Threshold 2 Temperature (THRESHOLD2_TEMP):** Thermal interrupt threshold temperature in degrees Celsius. Described in a signed, 2's complement format with the LSB representing 1'C resolution (S8.7.0). E.g., a reading of 0x28 == 40'C. This threshold is managed relative to the filtered temperature.
15:8	0h RW	**Thermal Threshold 1 Temperature (THRESHOLD1_TEMP):** Thermal interrupt threshold temperature in degrees Celsius. Described in a signed, 2's complement format with the LSB representing 1'C resolution (S8.7.0). E.g., a reading of 0x28 == 40'C. This threshold is managed relative to the filtered temperature.
7:3	0h RO	**Reserved (RSVD):** Reserved
2	0h RW	**Critical Thermal Event Interrupt Enable (CRITICAL_THERMAL_INT_ENABLE):** Enable thermal interrupt generation when the processor has detected a critical thermal event that requires immediate servicing. This event is intended to be an alert indicating thermal control failure and is an early warning of thermal runaway.
1	0h RW	**THRESHOLD2_INT_ENABLE:** When set, enables the generation of a thermal interrupt whenever the Thermal Threshold 2 Temperature is crossed. Interrupts are generated when the threshold is crossed in either direction. Interrupt destination is programmed in the TMBAR configuration space.
0	0h RW	**Threshold 1 Interrupt Enable (THRESHOLD1_INT_ENABLE):** When set, enables the generation of a thermal interrupt whenever the Thermal Threshold 1 Temperature is crossed. Interrupts are generated when the threshold is crossed in either direction. Interrupt destination is programmed in the TMBAR configuration space.

3.489 Package Thermal Status (P_CR_PKG_THERM_STATUS_0_0_0_MCHBAR) — Offset 701Ch

This register is used to monitor the status of the package level virtual thermal sensor and details on the source of package level thermal events. The package level virtual thermal sensor is a filtered version of the mximum temperature observed at any domain within the package. That temperture is applied to the Package Thermal Interrupt configuration for event delivery to software for run-time thermal management. When an event is observed, this register describe the source(s) of that event.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 701Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23:16	0h RO/V	**TEMPERATURE:** Virtual maximum SOC temperature in degrees Celsius. Caculated as a maximum of all on-die thermal sensor readings and filtered accoring to the time constant described in the PKG_THERM_INTERRUPT register. Data format is signed, 2's complement with the LSB representing 1'C resolution (S8.7.0).
15:6	0h RO	**Reserved (RSVD):** Reserved
5	0h RW/0C/V	**Critical Thermal Event Log (CRITICAL_THERMAL_EVENT_LOG):** Sticky log bit indicating that the processor has operated out of its thermal specification since the last time software cleared this bit. Set by hardware on a 0 to 1 transition of Critical Thermal Event Status.
4	0h RO/V	**Critical Thermal Event Status (CRITICAL_THERMAL_EVENT_STATUS):** Status bit indicating that the processor is operating outside of its thermal specification. It is intended as an early warning of thermal runaway in the silicon and shutdown is recommended.
3	0h RW/0C/V	**Thermal Threshold 2 Log (THRESHOLD2_LOG):** Sticky log bit that indicates temperature has crossed the software programmable thermal threshold2 in either falling or rising directions.
2	0h RO/V	**Thermal Threshold 2 Status (THRESHOLD2_STATUS):** Indicates that the current filtered temperature (bits 23:16 of this register) is greater than or equal to the Threshold2 defined in the PKG_THERM_INTERRUPT configuration register.
1	0h RW/0C/V	**Thermal Threshold 1 Log (THRESHOLD1_LOG):** Sticky log bit that indicates temperature has crossed the software programmable thermal threshold1 in either falling or rising directions.
0	0h RO/V	**Thermal Threshold 1 Status (THRESHOLD1_STATUS):** Indicates that the current filtered temperature (bits 23:16 of this register) is greater than or equal to the Threshold1 defined in the PKG_THERM_INTERRUPT configuration register.

3.490 LPDDR DRAM Thermal (MR4) Status of Channel 01 (P_CR_MEM_MR4_TEMPERATURE_DEV1_0_0_0_MCHBAR) — Offset 7024h

LPDDR DRAM Thermal (MR4) Status of Channel 01, when there are multiple DRAMs in a rank, the maximun MR4 is reported

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7024h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RO	**RESERVED_0:** Reserved
5:3	0h RW	**MR4 DRAM thermal status of Channel 01 Rank 1 (MR4_RANK_1):** This field is updated each read of LPDDR DRAM MR4 Device Temperature Status per rank. Update rate is configured by BIOS.
2:0	0h RW	**MR4 DRAM thermal status of Channel 01 Rank 0 (MR4_RANK_0):** This field is updated each read of LPDDR DRAM MR4 Device Temperature Status per rank. Update rate is configured by BIOS.

3.491 LPDDR DRAM Thermal (MR4) Status of Channel 10 (P_CR_MEM_MR4_TEMPERATURE_DEV2_0_0_0_MCHBAR) — Offset 7028h

LPDDR DRAM Thermal (MR4) Status of Channel 10, when there are multiple DRAMs in a rank, the maximun MR4 is reported

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7028h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RO	**RESERVED_0:** Reserved
5:3	0h RW	**MR4 DRAM thermal status of Channel 10 Rank 1 (MR4_RANK_1):** This field is updated each read of LPDDR DRAM MR4 Device Temperature Status per rank. Update rate is configured by BIOS.
2:0	0h RW	**MR4 DRAM thermal status of Channel 10 Rank 0 (MR4_RANK_0):** This field is updated each read of LPDDR DRAM MR4 Device Temperature Status per rank. Update rate is configured by BIOS.

3.492 Machine Check Error Source Log (P_CR_MCA_ERROR_SRC_0_0_0_MCHBAR) — Offset 702Ch

This register logs error source information i.e IERR or MCERR information for Pcode. The error fields are cleared by HW or BIOS. This register is also shadowed in the I/O space.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 702Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/V	**CATERR:** Asserted by HW on IERR or MCERR assertion.
30	0h RW/V	**IERR:** Asserted by HW on IERR assertion.
29	0h RW/V	**MCERR:** Asserted by HW on MCERR assertion.
28:8	0h RO	**Reserved (RSVD):** Reserved
7:0	0h RW	**RESERVED:** Reserved

3.493 DDR Thermal Throttling Control (P_CR_DDR_THERM_THRT_CTRL_0_0_0_MCHBAR) — Offset 7030h

This register is used to configure thermal throttling policies for memory.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7030h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**RESERVED_1:** Reserved
25	0h RW/V	**DDR3L Refresh Rate (DDR3L_REFRESH_RATE):** This field is to allow platform software to request refresh rate for DDR3L
24	0h RW/V	**DDR3L Throttle Enable (THROTTLE_LEVEL_ENABLE):** When set and DRAM_Type is DDR3L, SOC throttles memory traffic to level specified in THRT_LVL NOTE: This field is ignored if DRAM_Type is not DDR3L
23:16	0h RW/V	**DDR3L Throttle Level (THROTTLE_LEVEL):** Throttle level in %BW, in units of 1%. Default = 0%. Input of 100 or higher is clipped to 100%, NOTE: this field is ignored if DRAM_type is not DDR3L
15:10	0h RO	**RESERVED_0:** Reserved
9	0h RW/V	**Memory Thermal Throttle Enable (MEM_THRT_ENABLE):** When set, memory traffic is throttled if memory MR4 value >= THERM_THRT_THRESHOLD, respectively. Thermal throttling is achieved by applying memory bandwidth clips in the memory subsystem.
8	0h RW/V	**Memory Thermal Throttling Configuration (MEM_THRT_CFG):** Configure memory throttling behavior. Policies are defined as follows: • 0 = Thermal throttling policy uses instanous MR4 status for THERM_THRT_THRESHOLD • 1 = Thermal throttling policy uses time filtered MR4 status for THERM_THRT_THRESHOLD. Filtering time constant is configured in the DDR_THERM_INTERRUPT register.
7	0h RW/V	**Reserved (NM_THERM_THRT_ENABLE):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
6:4	0h RW/V	**Reserved (NM_THERM_THRT_THRESHOLD):** Reserved
3	0h RW/V	**LPDDR Memory Thermal Throttling Enable (FM_THERM_THRT_ENABLE):** When set, throttling is activated if LPDDR memory MR4 value is greater than or equal to THERM_THRT_THRESHOLD. For the standard LPDDR DRAM that is only capable up to Tcasemax of 85C this bit should be set to enable LPDDR throttling to keep DRAM within its Tcasemax spec.
2:0	0h RW/V	**LPDDR Memory Thermal Throttling Threshold (FM_THERM_THRT_THRESHOLD):** Configurable threshold of LPDDR memory MR4 value greater than or equal to which thermal throttling is activated. For the standard LPDDR DRAM that is only capable up to Tcasemax of 85C this field should be configured to avoid DRAM to exceed its Tcasemax spec. • Memory MR4 >= Threshold = Enable thermal throttling • Memory MR4 < Threshold = Disable thermal throttling

3.494 DDR Thermal Interrupt Control (P_CR_DDR_THERM_INTERRUPT_0_0_0_MCHBAR) — Offset 7034h

This register is used to manage DDR thermal interrupts, including management of filtering on the virtual DRAM thermal sensor control signal. These features are designed to allow software to implement smooth control of thermally significant events for platform thermal management.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7034h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved
30:24	0h RW	**Time Window (TIME_WINDOW):** Virtual Temperature thermal filter RC time constant. Virtual temperature readings are run through an RC filter before they are fed into status and interrupt generation. This filtering allows software smooth control of thermal responses to thermally significant events. The bits of this field describe parameters for a mathematical equation for time window configuration. This field is split into two sub-fields: • x = bits[6:5] • y = bits[4:0] Time window equation: **time_window = PACKAGE_POWER_SKU_UNIT.TIME_UNIT * ((1+x/4)^y)**
23:16	0h RO	**Reserved (RSVD):** Reserved
15	0h RW	**Reserved (NEAR_MEM_MR4_THRESHOLD2_INT_ENABLE):** Reserved
14:12	0h RW	**Reserved (NEAR_MEM_MR4_THRESHOLD2):** Reserved
11	0h RW	**Reserved (NEAR_MEM_MR4_THRESHOLD1_INT_ENABLE):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
10:8	0h RW	**Reserved (NEAR_MEM_MR4_THRESHOLD1):** Reserved
7	0h RW	**Memory MR4 Threshold 2 Interrupt Enable (FAR_MEM_MR4_THRESHOLD2_INT_ENABLE):** Enable thermal interrupt generation whenever the virtual maximum memory MR4 has crossed THRESHOLD2. Interrupts are triggered when the filtered MR4 temperature crosses in both rising and falling directions.
6:4	0h RW	**Memory MR4 Threshold 2 (FAR_MEM_MR4_THRESHOLD2):** Configurable memory threshold2 value for memory thermal interrupt generation
3	0h RW	**Memory MR4 Threshold 1 Interrupt Enable (FAR_MEM_MR4_THRESHOLD1_INT_ENABLE):** Enable thermal interrupt generation whenever the virtual maximum memory MR4 has crossed THRESHOLD1. Interrupts are triggered when the filtered MR4 temperature crosses in both rising and falling directions.
2:0	0h RW	**Memory MR4 Threshold 1 (FAR_MEM_MR4_THRESHOLD1):** Configurable memory threshold1 value for memory thermal interrupt generation

3.495 DDR Thermal Status (P_CR_DDR_THERM_STATUS_0_0_0_MCHBAR) — Offset 7038h

Status register for monitoring DDR thermal status. Data reported here is aggregated by memory type, and values reported represent the maximum MR4 readings observed in those respective domains. Temperatures are additionally filtered by the MR4 thermal filtering time constant described in the DDR Thermal Interrupt configuration register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7038h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (RSVD):** Reserved
21:19	0h RO/V	**Reserved (NEAR_MEM_MR4):** Reserved
18:16	0h RO/V	**LPDDR Memory MR4 (FAR_MEM_MR4):** Virtual maximum memory MR4 reading. Caculated as a maximum of all LPDDR memory MR4 readingsand filtered accoring to the time constant described in DDR_THERM_INTERRUPT register.
15:8	0h RO	**Reserved (RSVD):** Reserved
7	0h RW/0C/V	**Reserved (NEAR_MEM_MR4_THRESHOLD2_LOG):** Reserved
6	0h RO/V	**Reserved (NEAR_MEM_MR4_THRESHOLD2_STATUS):** Reserved
5	0h RW/0C/V	**Reserved (NEAR_MEM_MR4_THRESHOLD1_LOG):** Reserved
4	0h RO/V	**Reserved (NEAR_MEM_MR4_THRESHOLD1_STATUS):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW/0C/V	**Memory MR4 Threshold2 Log (FAR_MEM_MR4_THRESHOLD2_LOG):** Indicates that the virtual maximum memory MR4 has crossed THRESHOLD2 since the last time this register was cleared. The bit is set when the threshold is crossed in either direction. Software may clear this bit.
2	0h RO/V	**Memory MR4 Threshold2 Status (FAR_MEM_MR4_THRESHOLD2_STATUS):** Status bit is set when the virtual maximum memory MR4 reading is greater than or equal to THRESHOLD2. It is cleared when temperature is less than THRESHOLD2
1	0h RW/0C/V	**Memory MR4 Threshold1 Log (FAR_MEM_MR4_THRESHOLD1_LOG):** Indicates that the virtual maximum memory MR4 has crossed THRESHOLD1 since the last time this register was cleared. The bit is set when the threshold is crossed in either direction. Software may clear this bit.
0	0h RO/V	**Memory MR4 Threshold1 Status (FAR_MEM_MR4_THRESHOLD1_STATUS):** Status bit is set when the virtual maximum memory MR4 reading is greater than or equal to THRESHOLD1. It is cleared when temperature is less than THRESHOLD1

3.496 Dram Energy Counter (P_CR_DDR_ENERGY_STATUS_0_0_0_MCHBAR) — Offset 7048h

Reports total energy consumed in DRAM. The energy status is reported in units which are defined by PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT. The counter will wrap around and continue counting from zero when it reaches its limit and therefore should be polled sufficiently frequently to avoid aliasing. Typically, software will calculate delta energy and delta time and divide the two to estimate Watts consumed over a time window. The value of this register is updated at approximately every 1ms. This energy status is what is used by DDR RAPL or OLTM control algorithms if the product supports those features. To calculate Watts: **Watts = delta(energy) / delta(time) / 2^PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT** THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7048h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**JOULES:** Total Joules of energy consumed by all DIMMs. Units are proportional to Joules and are defined by PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT.

3.497 DDR RAPL Performance Status (P_CR_DDR_RAPL_PERF_STATUS_0_0_0_MCHBAR) — Offset 704Ch

Memory RAPL performance throttling counter. This counter accumulates time that any channel in the memory controller is bandwidth throttled due to memory RAPL constraints. This counter counts total time (in PACKAGE_POWER_SKU_UNIT_MSR.TIME_UNIT units) that any channel is throttled. If two channels are throttled, this counter increments at a 2x rate, so that for 1ms in wall clock time the counter counts 2ms. This counter does not include throttling as a result

of thermal management or MEMHOT. This register is updated at approximately 1ms intervals. This counter is normalized to 'seconds' and is not subject to variation of actual DRAM clock speeds. This register is readonly for software via MMIO MSR and PECI/PCS. This register starts counting at zero from reset and continues counting forever and wraparounds may occur, so software should ensure the sample rate is sufficient to avoid aliasing. This is an unsigned value.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 704Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DDR Bandwidth Throttle Duration (DURATION):** Bandwidth throttle duration counter due to Memory RAPL. Sum across all channels in PACKAGE_POWER_SKU_UNIT_MSR.TIME_UNIT units. This data can serve as a proxy for the potential performance impacts of RAPL on memory accesses.

3.498 Package RAPL Performance Status (P_CR_PACKAGE_RAPL_PERF_STATUS_0_0_0_MCHBAR) — Offset 7050h

Counts time that any core in the IA domain is performance throttled below OS request and below the base frequency (P1) because of power limits (PL1 or PL2). Counts in time units defined by PACKAGE_POWER_SKU_UNIT_MSR.TIME_UNIT. If software uses the TURBO_ACTIVATION_RATIO or PECI ACPI P-NOTIFY, the turbo activation ratios described by those features will may elevate the effective OS request (as calculated by this counter) to max turbo. This register starts counting at zero from reset and continues counting forever and wraparounds may occur, so software should ensure the sample rate is sufficient to avoid aliasing. This is an unsigned value.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7050h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Performance Throttle Duration (COUNTS):** Time that any core in the IA domain is performance throttled below OS request and below the base frequency (P1) because of power limits (PL1 or PL2). Counts in time units defined by PACKAGE_POWER_SKU_UNIT_MSR.TIME_UNIT

3.499 IA Core Performance / Power Priority Control (P_CR_PRIMARY_PLANE_TURBO_PLCY_0_0_0_MCHBAR) — Offset 7054h

The PRIMARY_PLANE_TURBO_POWER_POLICY and SECONDARY_PLANE_TURBO_POWER_POLICY are used together as hints to balance the power budget between the primary (IA core) and secondary (Graphics) power planes. This biasing is effectively a performance biasing, and it helps Punit firmware assess where software needs performance the most

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7054h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RO	**RESERVED_0:** Reserved
4:0	0h RW	**IA Core Priority Level (PRIPTP):** Performance priority Level for the IA Core (primary) power plane. A higher number implies a higher priority.

3.500 Graphics Performance / Power Priority Control (P_CR_SECONDARY_PLANE_TURBO_PLCY_0_0_0_MCHBAR) — Offset 7058h

The PRIMARY_PLANE_TURBO_POWER_POLICY and SECONDARY_PLANE_TURBO_POWER_POLICY are used together as hints to balance the power budget between the primary (IA core) and secondary (Graphics) power planes. This biasing is effectively a performance biasing, and it helps Punit firmware assess where software needs performance the most

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7058h	10 h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RO	**RESERVED_0:** Reserved
4:0	10h RW	**Graphics Priority Level (SECPTP):** Performance priority Level for the Graphics (secondary) power plane. A higher number implies a higher priority.

3.501 IA Energy Counter (P_CR_PRIMARY_PLANE_ENERGY_STATUS_0_0_0_MCHBAR) — Offset 705Ch

Reports total energy consumed across all IA cores. The energy status is reported in units which are defined by PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT. The counter will wrap around and continue counting from zero when it reaches its limit and therefore should be polled sufficiently frequently to avoid aliasing. Typically, software will calculate delta energy and delta time and divide the two to estimate Watts consumed over a time window. The value of this register is updated at approximately every 1ms. To calculate Watts: **Watts = delta(energy) / delta(time) / 2^PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT** THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 705Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**IA Energy Counter (DATA):** Contains an accumulated value of the energy consumed in the primary power plane. To find the energy consumed in a given time window, sofwtare should subtract the two energy readings. Software will have to take care of counter wrapping around when it overflows. Units are proportional to Joules exact precsion is defined by PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT

3.502 Graphics Energy Counter (P_CR_SECONDARY_PLANE_ENERGY_STATUS_0_0_0_MCHBAR) — Offset 7060h

Reports total energy consumed across all IA cores. The energy status is reported in units which are defined by PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT. The counter will wrap around and continue counting from zero when it reaches its limit and therefore should be polled sufficiently frequently to avoid aliasing. Typically, software will calculate delta energy and delta time and divide the two to estimate Watts consumed over a time window. The value of this register is updated at approximately every 1ms. To calculate Watts: **Watts = delta(energy) / delta(time) / 2^PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT** THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7060h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Graphics Energy Counter (DATA):** Contains an accumulated value of the energy consumed in the secondary power plane. To find the energy consumed in a given time window, sofwtare should subtract the two energy readings. Software will have to take care of counter wrapping around when it overflows. Units are proportional to Joules exact precsion is defined by PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT

3.503 PACKAGE_POWER_SKU_UNIT (P_CR_PACKAGE_POWER_SKU_UNIT_0_0_0_MCHBAR) — Offset 7068h

Defines units for calculating SKU power, current, energy, resistance and timing parameters.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7068h	330A0E08 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	3h RW	**RESISTANCE_UNIT:** Used to define the units of resistance for control registers that describe parameters in ohms such as VR_CURRENT_CONFIG. The actual unit value is calculated by 1mohm / 2^RESISTANCE_UNIT. The default value of 3 corresponds to 0.125mohm.
27:24	3h RW	**CURRENT_UNIT:** Used to define the units of amps in control registers such as VR_CURRENT_CONFIG. The actual unit value is calculated by 1A / 2^CURRENT_UNIT. The default value of 3 corresponds to 0.125A.
23:20	0h RSV	**RESERVED_2:** Reserved
19:16	Ah RW	**TIME_UNIT:** Used for to define the time units in registers such as PL1, PL2, PL3 and PL4. The actual unit value is calculated by 1s / 2^TIME_UNIT. The default value of 10 corresponds to 0.977ms.
15:13	0h RSV	**RESERVED_1:** Reserved
12:8	Eh RW	**ENERGY_UNIT:** Used to define the units of energy reporting registers such as PACKAGE_ENERGY_STATUS. The actual unit value is calculated by 1 J / 2^ENERGY_UNIT. The default value of 14 corresponds to ~61uJ per bit.
7:4	0h RSV	**RESERVED_0:** Reserved
3:0	8h RW	**PWR_UNIT:** Used to define the units of power control registers such as PL1, PL2, PL3 and PL4. The actual unit value is calculated by 1 W / 2^PWR_UNIT. The default value of 8 corresponds to 3.9mW per bit.

3.504 SOC Energy Counter (P_CR_PACKAGE_ENERGY_STATUS_0_0_0_MCHBAR) — Offset 706Ch

Reports total energy consumed across the entire SOC / Package. The energy status is reported in units which are defined by PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT. The counter will wrap around and continue counting from zero when it reaches its limit and therefore should be polled sufficiently frequently to avoid aliasing. Typically, software will calculate delta energy and delta time and divide the two to estimate Watts consumed over a time window. The value of this register is updated at approximately every 1ms. This energy status is what is used by RAPL PL1, PL2 and PL3 control algorithms. To calculate Watts: **Watts = delta(energy) / delta(time) / 2^PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT** THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 706Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**SOC Energy Counter (DATA):** Contains accumulated energy consumed by the entire CPU. This counter will wrap around and keep counting when the counter overflows. Units are proportional to Joules exact precsion is defined by PACKAGE_POWER_SKU_UNIT_MSR.ENERGY_UNIT

3.505 GT_PERF_STATUS (P_CR_GT_PERF_STATUS_0_0_0_MCHBAR) — Offset 7070h

Contains the voltage and ratio status for GT. This register is mapped to GT_PERF_STATUS_0_0_0_MCHBAR.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7070h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (RSVD):** Reserved
25:17	0h RO/V	**RP_STATE_RATIO_SLICE:** Ratio of the current RP-state, in 16.6Mhz 1xclks. When the graphics engine is in RC6, this field is zeroed out.
16:8	0h RO/V	**RP_STATE_RATIO_UNSLICE:** Ratio of the current RP-state, in 16.6Mhz 1xclks. When the graphics engine is in RC6, this field is zeroed out.
7:0	0h RO/V	**RP_STATE_VOLTAGE:** RP-State Voltage GT Target Voltage in U1.7 Volts

3.506 Temperature Reference and Control (P_CR_TEMPERATURE_TARGET_0_0_0_MCHBAR) — Offset 7074h

This register contains information about the fan speed control target temperature as well as details on the reference temperature for IA core DTS relative temperature reading. **MSR_Name**: TEMPERATURE_TARGET **MSR_Addr**: 0x1A2

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7074h	5A0000 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved
30:24	0h RW	**TJ_MAX_TCC_OFFSET:** This field allows platform software to configure the temperature at which thermal monitor engages to be lower than the manufacturing configured maximum constraint. This field is programmed in 1'C units. E.g., if the default silicon configured maximum temperature is 100'C and this field is configured to 10, then the silicon will engage thermal throttling algorithms at 90'C
23:16	5Ah RO_V	**REF_TEMP:** Tjmax a.k.a. Thermal Monitor activation temperature or Prochot Temperature. This is the maximum junction temperature at which thermal throttling aka thermal monitor is activated. This temperature is the maximum temperature at which the silicon is capable of operating at. All IA core digital thermal sensor readings are reported as a relative nagative offset from this reference temperature, such that a readon of zero implies the cores are running at this temperature.
15:8	0h RO/V	**Fan Temperature Target Offset (FAN_TEMP_TARGET_OFFSET):** Fan Temperature Target Offset a.k.a. TControl indicates the relative offset from the the Thermal Monitor Trip Temperature at which fans should be engaged.
7:0	0h RO	**Reserved (RSVD):** Reserved

3.507 BIOS Reset Completion (P_CR_BIOS_RESET_CPL_0_0_0_MCHBAR) — Offset 7078h

This register is used as a means for BIOS to communicate staging to the Punit / Pcode. The exact definition and utility of each bit may differ across products. The general philosophy is that when BIOS is done with stage0, it writes the RST_CPL0 bit and then waits for the PCODE_INIT_DONE0 bit to be set before proceeding to the next step.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7078h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**RESERVED0:** Reserved
15	0h RO/V	**Stage7 Pcode Reset Complete (PCODE_INIT_DONE7):** Pcode sets this bit when it has completed this stage, BIOS must wait for this bit to be set before proceeding to the next stage. Stage validity is product-specific and this stage may not be applicable to this product.
14	0h RO/V	**Stage6 Pcode Reset Complete (PCODE_INIT_DONE6):** Pcode sets this bit when it has completed this stage, BIOS must wait for this bit to be set before proceeding to the next stage. Stage validity is product-specific and this stage may not be applicable to this product.
13	0h RO/V	**Stage5 Pcode Reset Complete (PCODE_INIT_DONE5):** Pcode sets this bit when it has completed this stage, BIOS must wait for this bit to be set before proceeding to the next stage. Stage validity is product-specific and this stage may not be applicable to this product.
12	0h RO/V	**Stage4 Pcode Reset Complete (PCODE_INIT_DONE4):** Pcode sets this bit when it has completed this stage, BIOS must wait for this bit to be set before proceeding to the next stage. Stage validity is product-specific and this stage may not be applicable to this product.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RO/V	**Stage3 Pcode Reset Complete (PCODE_INIT_DONE3):** Pcode sets this bit when it has completed this stage, BIOS must wait for this bit to be set before proceeding to the next stage. Stage validity is product-specific and this stage may not be applicable to this product.
10	0h RO/V	**Stage2 Pcode Reset Complete (PCODE_INIT_DONE2):** Pcode sets this bit when it has completed this stage, BIOS must wait for this bit to be set before proceeding to the next stage. Stage validity is product-specific and this stage may not be applicable to this product.
9	0h RO/V	**Stage1 Pcode Reset Complete (PCODE_INIT_DONE1):** Pcode sets this bit when it has completed this stage, BIOS must wait for this bit to be set before proceeding to the next stage. Stage validity is product-specific and this stage may not be applicable to this product.
8	0h RO/V	**Stage0 Pcode Reset Complete (PCODE_INIT_DONE):** Pcode has completed its actions in response to Stage0 BIOS Reset complete. Between BIOS Stage0 complete and pcode Stage0 complete, pcode will apply all power savings configurations to PCS and will set up C_STATE_LATENCY control MSR settings for IRTL management.
7	0h RW	**Stage7 BIOS Reset Complete (RST_CPL7):** reset complete
6	0h RW	**Stage6 BIOS Reset Complete (RST_CPL6):** BIOS sets this bit when it has completed this stage, Pcode must wait for this bit to be set before proceeding to the next stage. Stage validity is product-specific and this stage may not be applicable to this product.
5	0h RW	**Stage5 BIOS Reset Complete (RST_CPL5):** BIOS sets this bit when it has completed this stage, Pcode must wait for this bit to be set before proceeding to the next stage. Stage validity is product-specific and this stage may not be applicable to this product.
4	0h RW	**Stage4 BIOS Reset Complete (RST_CPL4):** BIOS sets this bit when it has completed this stage, Pcode must wait for this bit to be set before proceeding to the next stage. Stage validity is product-specific and this stage may not be applicable to this product.
3	0h RW	**Stage3 BIOS Reset Complete (RST_CPL3):** BIOS sets this bit when it has completed this stage, Pcode must wait for this bit to be set before proceeding to the next stage. Stage validity is product-specific and this stage may not be applicable to this product.
2	0h RW	**Stage2 BIOS Reset Complete (RST_CPL2):** BIOS sets this bit when it has completed this stage, Pcode must wait for this bit to be set before proceeding to the next stage. Stage validity is product-specific and this stage may not be applicable to this product.
1	0h RW	**Stage1 BIOS Reset Complete (RST_CPL1):** BIOS sets this bit when it has completed this stage, Pcode must wait for this bit to be set before proceeding to the next stage. Stage validity is product-specific and this stage may not be applicable to this product.
0	0h RW	**Stage0 BIOS Reset Complete (RST_CPL):** Set by BIOS to indicate that all power management configurations as part of reset are complete. This must include Punit patch load done as well as all relevant Punit power management register and mailbox configurations done. Once this bit is set, Punit will allow normal power management to start. Before setting this bit, P-states and C-states support is disabled. BIOS should wait before receiving the Pcode Stage0 reset complete before proceeding with any further steps.

3.508 BIOS_MAILBOX_DATA (P_CR_BIOS_MAILBOX_DATA_0_0_0_MCHBAR) — Offset 7080h

Data register for the BIOStoPCODE mailbox. This mailbox is implemented as a means for accessing statistics and implementing BIOS-Pcode handshakes. This register is used in conjunction with BIOS_MAILBOX_INTERFACE. THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7080h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**DATA:** This field contains the data associated with specific commands.

3.509 BIOS_MAILBOX_INTERFACE (P_CR_BIOS_MAILBOX_INTERFACE_0_0_0_MCHBAR) — Offset 7084h

Control and Status register for the BIOStoPCODE mailbox. This mailbox is implemented as a means for accessing statistics and implementing BIOS-Pcode handshakes. This register is used in conjunction with BIOS_MAILBOX_DATA. THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7084h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S/V	**RUN_BUSY:** SW may write to the two mailbox registers only when RUN_BUSY is clear(0). Setting RUN_BUSY to 1 will pend a Fast Path event to Pcode. After setting this bit SW will poll this bit until it is cleared. PCODE will clear RUN_BUSY after updating the mailbox registers with the result and error code.
30:29	0h RO	**Reserved (RSVD):** Reserved
28:8	0h RW/V	**ADDRESS:** This field is used to specify an additional parameter to extend the command when needed.
7:0	0h RW/V	**COMMAND:** This field contains the SW request command or the PCODE response code depending on the setting of RUN_BUSY.

3.510 CORE_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_CORE_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR) — Offset 7088h

PUNIT_MMIO: Core Frequency Capabilities This register describes the frequency capabilities of the IA cores. Units are 100MHz multiplied by the ratio. Minimum and maximum ratio fields are initialized by pCode at reset. Last resolved ratio is updated upon changes to the processing system frequency. The efficient ratio is determined by firmware and may be updated dynamically depending on firmware support.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7088h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO/V	**LAST_RESOLVED_FREQ:** Last resolved ratio for the IA cores. Units are 100MHz multiplied by the ratio. This value is updated dynamically whenever the IA core frequency changes.
23:16	0h RO/V	**MAX_SUPPORTED_FREQ:** Maximum ratio for the IA cores. Units are 100MHz multiplied by the ratio.
15:8	0h RO/V	**EFFICIENT_FREQ:** Firmware-calculated efficient ratio for the IA cores. Units are 100MHz multiplied by the ratio.
7:0	0h RO/V	**MIN_SUPPORTED_FREQ:** Minimum supported ratio for the IA cores. Units are 100MHz multiplied by the ratio.

3.511 GRAPHICS_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_GRAPHICS_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR) — Offset 708Ch

PUNIT_MMIO: Graphics Engine Frequency Capabilities This register describes the frequency capabilities of the integrated graphics engine. Units are 16.67MHz multiplied by the ratio. Minimum and maximum ratio fields are initialized by pCode at reset. Last resolved ratio is updated upon changes to the integrated graphics engine frequency. The efficient ratio is determined by firmware and may be updated dynamically depending on firmware support.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 708Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO/V	**LAST_RESOLVED_FREQ:** Last resolved ratio for the integrated graphics engine. Units are 16.67MHz multiplied by the ratio. This value is updated dynamically whenever the graphics engine frequency changes.
23:16	0h RO/V	**MAX_SUPPORTED_FREQ:** Maximum supported ratio for the integrated graphics engine. Units are 16.67MHz multiplied by the ratio.
15:8	0h RO/V	**EFFICIENT_FREQ:** Firmware-calculated efficient ratio for the integrated graphics engine. Units are 16.67MHz multiplied by the ratio.
7:0	0h RO/V	**MIN_SUPPORTED_FREQ:** Minimum supported ratio for the integrated graphics engine. Units are 16.67MHz multiplied by the ratio.

3.512 SYSTEM_AGENT_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR

(P_CR_SYSTEM_AGENT_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR) — Offset 7090h

PUNIT_MMIO: System Agent Frequency Capabilities This register describes the frequency capabilities of the System Agent. Units are 16.666MHz multiplied by the ratio. Last resolved ratio is updated upon changes to the System Agent frequency.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7090h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO/V	**LAST_RESOLVED_RATIO:** Last resolved System Agent ratio, in units of 16.666MHz.
23:16	0h RO/V	**MAX_RATIO:** max ratio
15:8	0h RO/V	**RESERVED_1:** Reserved
7:0	0h RO/V	**MIN_RATIO:** min ratio

3.513 Memory Frequency Status (P_CR_FAR_MEMORY_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR) — Offset 7094h

This register reports out the LPDDR memory frequency. The actual capabilities of the SOC with respect to LPDDR frequency is described in the MEMSS_FREQUENCY_CAPABILITIES register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7094h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO/V	**Last Resolved Memory Frequency (LAST_RESOLVED_RATIO):** This field reports out the LPDDR memory frequency in integer multiple of 133.33MHz. This register reflects what BIOS has programmed as the default LPDDR frequency in products that do not support run-time memory frequency control. For products supporting run-time memory frequency control, this field describes the last resolved frequency.
23:16	0h RO/V	**RESERVED_2:** Reserved
15:8	0h RO/V	**RESERVED_1:** Reserved
7:0	0h RO/V	**RESERVED_0:** Reserved

3.514 Package Power SKU and RAPL Power Control Capabilities (P_CR_PACKAGE_POWER_SKU_0_0_0_MCHBAR) — Offset 70A0h

This register describes the the power SKU of the part and limits on time window and power limit configuration allowed in the RAPL PL1 and PL2 configuration registers.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 70A0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:55	0h RO	**RESERVED_3:** Reserved
54:48	0h RW	**RAPL Maximum Allowed Time Window (PKG_MAX_WIN):** The maximal time window allowed to be programmed for RAPL PL1 and PL2 controls for the SKU. Higher values will be clamped to this value. The bits of this field describe parameters for a mathematical equation for time window configuration. This field is split into two sub-fields: • x = bits[6:5] • y = bits[4:0] Time window equation: **time_window = PACKAGE_POWER_SKU_UNIT.TIME_UNIT * ((1+x/4)^y)**
47	0h RO	**RESERVED_2:** Reserved
46:32	0h RW	**RAPL Maximum Power Limit (PKG_MAX_PWR):** The maximal package power setting allowed for the SKU. Higher values will be clamped to this value. The maximum setting is typical not guaranteed. The default value for this field is determined by fuses. The units for this value are defined in PACKAGE_POWER_SKU_MSR[PWR_UNIT].
31	0h RO	**RESERVED_1:** Reserved
30:16	0h RW	**Package Minimum Power (PKG_MIN_PWR):** The minimal package power setting allowed for the SKU. Lower values may not be achievable by run-time RAPL PL1 and PL2 control algorithms.
15	0h RO	**RESERVED_0:** Reserved
14:0	0h RW	**PKG_TDP:** The TDP package power setting allowed for the SKU. The TDP setting is typical not guaranteed. The default value for this field is determined by fuses. The units for this value are defined in PACKAGE_POWER_SKU_MSR[PWR_UNIT].

3.515 Package RAPL Power Limit (P_CR_PACKAGE_RAPL_LIMIT_0_0_0_MCHBAR) — Offset 70A8h

Package RAPL Power Limit allows a software agent to define power limitation for the package domain. Power limitation is defined in terms of average power usage (Watts) over a time window specified. Two power limits and associated time windows can be specified. These power limits are commonly referred to as PL1 (long time window) and PL2 (short time window). Each power limit provides independent clamping control that would permit the processor cores to go below OS-requested state to meet the power

limits. A lock mechanism allow the software agent to enforce power limit settings. Once the lock bit is set, the power limit settings are static and un-modifiable until next RESET.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 70A8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW/L	**Package RAPL Lock (PKG_PWR_LIM_LOCK):** When set all settings in this register are locked and are treated as Read Only. This lock control is persistent until the next reset. This bit will typically set by BIOS during boot time or resume from Sx.
62:56	0h RSV	**RESERVED_1:** Reserved
55:49	0h RW/L	**Power Limit 2 (PL2) Time Window (PKG_PWR_LIM_2_TIME):** Time window for Power Limit 1 (PL2). This describes the control window of the power limit. This time window is described in an RC time constant format, which means that if 1s is programmed, the power limit constraint really applies at more like 5s. The maximal time window is bounded by PACKAGE_POWER_SKU_MSR.PKG_MAX_WIN. Thre is no constraint on the minimum programmable time window, however at very short time windows the control algorithms may not be effective. The bits of this field describe parameters for a mathematical equation for time window configuration. This field is split into two sub-fields: • x = bits[6:5] • y = bits[4:0] Time window equation: **time_window = PACKAGE_POWER_SKU_UNIT.TIME_UNIT * ((1+x/4)^y)**
48	0h RW/L	**Power Limit 2 (PL2) Clamp (PKG_CLMP_LIM_2):** Clamp mode control for PL2. • 0 = PL2 power control is prevented from forcing P-states below the base frequency / P1 for any domain in the SOC. • 1 = PL2 power control will take all actions necessary to meet the power target, even if that involves running at clock frequencies below the base frequency / P1 level. In order to ensure proper SOC cooling, it is generally recommended that the clamp mode is always enabled.
47	0h RW/L	**Power Limit 2 (PL2) Enable (PKG_PWR_LIM_2_EN):** Enable for Power Limit 2 (PL2). Setting this bit activates the power limit and time window defined for PL2.
46:32	0h RW/L	**Power Limit 2 (PL2) (PKG_PWR_LIM_2):** Sets the average power usage limit of the package domain corresponding to the PL2 time window. The power units of this field are specified by the PACKAGE_POWER_SKU_UNIT_MSR.PWR_UNIT. This power limit must be configured by software before it will engage. The PL2 limit is most commonly associated with long time windows (1s and longer), although there are no explicit constraints on what software configures.
31:24	0h RSV	**RESERVED_0:** Reserved
23:17	0h RW/L	**Power Limit 1 (PL1) Time Window (PKG_PWR_LIM_1_TIME):** Time window for Power Limit 1 (PL1). This describes the control window of the power limit. This time window is described in an RC time constant format, which means that if 1s is programmed, the power limit constraint really applies at more like 5s. The maximal time window is bounded by PACKAGE_POWER_SKU_MSR.PKG_MAX_WIN. Thre is no constraint on the minimum programmable time window, however at very short time windows the control algorithms may not be effective. The bits of this field describe parameters for a mathematical equation for time window configuration. This field is split into two sub-fields: • x = bits[6:5] • y = bits[4:0] Time window equation: **time_window = PACKAGE_POWER_SKU_UNIT.TIME_UNIT * ((1+x/4)^y)**

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW/L	**Power Limit 1 (PL1) Clamp (PKG_CLMP_LIM_1):** Clamp mode control for PL1. • 0 = PL1 power control is prevented from forcing P-states below the base frequency / P1 for any domain in the SOC. • 1 = PL1 power control will take all actions necessary to meet the power target, even if that involves running at clock frequencies below the base frequency / P1 level. In order to ensure proper SOC cooling, it is generally recommended that the clamp mode is always enabled.
15	0h RW/L	**Power Limit 1 (PL1) Enable (PKG_PWR_LIM_1_EN):** Enable for Power Limit 1 (PL1). Setting this bit activates the power limit and time window defined for PL1.
14:0	0h RW/L	**Power Limit 1 (PL1) (PKG_PWR_LIM_1):** Sets the average power usage limit of the package domain corresponding to the PL1 time window. The power units of this field are specified by the PACKAGE_POWER_SKU_UNIT_MSR.PWR_UNIT. This power limit must be configured by software before it will engage. The PL1 limit is most commonly associated with long time windows (1s and longer), although there are no explicit constraints on what software configures.

3.516 IA_PERF_LIMIT_REASONS (P_CR_IA_PERF_LIMIT_REASONS_0_0_0_MCHBAR) — Offset 70B0h

Interface to allow software to determine what is causing resolved frequency to be clamped below the requested frequency. Status bits are updated by Punit firmware every millisecond and log bits are set upon observing the corresponding status bit being set. Software may clear log bits to track performance limit reasons inbetween reads of this register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 70B0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/0C/V	**QOS_LOG:** Sticky log bit corresponding to the associated status bit in the lower 16 bits. Set by hardware when the status bit asserts. Cleared by software write of zero.
30	0h RW/0C/V	**MAX_EFFICIENCY_FREQ_LOG:** Sticky log bit corresponding to the associated status bit in the lower 16 bits. Set by hardware when the status bit asserts. Cleared by software write of zero.
29	0h RW/0C/V	**FREQUENCY_ATTENUATION_LOG (MCT_LOG):** Sticky log bit corresponding to the associated status bit in the lower 16 bits. Set by hardware when the status bit asserts. Cleared by software write of zero.
28	0h RW/0C/V	**EDP_LOG:** Sticky log bit corresponding to the associated status bit in the lower 16 bits. Set by hardware when the status bit asserts. Cleared by software write of zero.
27	0h RW/0C/V	**MAX_TURBO_LIMIT_LOG (MULTI_CORE_TURBO_LOG):** Sticky log bit corresponding to the associated status bit in the lower 16 bits. Set by hardware when the status bit asserts. Cleared by software write of zero.
26	0h RW/0C/V	**VR_THERMALERT_LOG:** Sticky log bit corresponding to the associated status bit in the lower 16 bits. Set by hardware when the status bit asserts. Cleared by software write of zero.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW/0C/V	**IA_UTILIZATION_LOG:** Sticky log bit corresponding to the associated status bit in the lower 16 bits. Set by hardware when the status bit asserts. Cleared by software write of zero.
24	0h RW/0C/V	**DEV3_LOG:** Sticky log bit corresponding to the associated status bit in the lower 16 bits. Set by hardware when the status bit asserts. Cleared by software write of zero.
23	0h RW/0C/V	**DEV2_LOG:** Sticky log bit corresponding to the associated status bit in the lower 16 bits. Set by hardware when the status bit asserts. Cleared by software write of zero.
22	0h RW/0C/V	**DEV3_POWER_THERMAL_LOG (SPARE6_LOG):** Sticky log bit corresponding to the associated status bit in the lower 16 bits. Set by hardware when the status bit asserts. Cleared by software write of zero.
21	0h RW/0C/V	**DEV2_POWER_THERMAL_LOG (SPARE5_LOG):** Sticky log bit corresponding to the associated status bit in the lower 16 bits. Set by hardware when the status bit asserts. Cleared by software write of zero.
20	0h RW/0C/V	**POWER_BALANCER_LOG (SPARE4_LOG):** Sticky log bit corresponding to the associated status bit in the lower 16 bits. Set by hardware when the status bit asserts. Cleared by software write of zero.
19	0h RW/0C/V	**PL2_LOG:** Sticky log bit corresponding to the associated status bit in the lower 16 bits. Set by hardware when the status bit asserts. Cleared by software write of zero.
18	0h RW/0C/V	**PL1_LOG:** Sticky log bit corresponding to the associated status bit in the lower 16 bits. Set by hardware when the status bit asserts. Cleared by software write of zero.
17	0h RW/0C/V	**THERMAL_LOG:** Sticky log bit corresponding to the associated status bit in the lower 16 bits. Set by hardware when the status bit asserts. Cleared by software write of zero.
16	0h RW/0C/V	**PROCHOT_LOG:** Sticky log bit corresponding to the associated status bit in the lower 16 bits. Set by hardware when the status bit asserts. Cleared by software write of zero.
15	0h RO/V	**QOS_STATUS:** Frequency is limited below the quality of service level.
14	0h RO/V	**MAX_EFFICIENCY_FREQ_STATUS:** Frequency is limited below the maxiumum efficiency operation point.
13	0h RO/V	**FREQUENCY_ATTENUATION_STATUS (MCT_STATUS):** Frequency is limited due frequency transition filtering algorithms. This most commonly occurs as a result of high frequency C-state transitions that result in icc-max or turbo ratio limits induced clock frequency transitions.
12	0h RO/V	**EDP_STATUS:** Frequency is limited due electrical data point constraints such as VR icc-max or battery max power limits
11	0h RO/V	**MAX_TURBO_LIMIT_STATUS (MULTI_CORE_TURBO_STATUS):** Frequency is limited due maximum allowed per-core turbo clock frequency constraints. These may be product specific limitations or software programmed limits.
10	0h RO/V	**VR_THERMALERT_STATUS:** Frequency is limited due to a VR thermal excursion.
9	0h RO/V	**IA_UTILIZATION_STATUS:** requency is limited due to autonomous utilization-based P-state control, if supported on Broxton
8	0h RO/V	**DEV3_STATUS:** Frequency is limited due to camera (device3) driver override.
7	0h RO/V	**DEV2_STATUS:** Frequency is limited due to graphics (device2) driver override.
6	0h RO/V	**DEV3_POWER_THERMAL_STATUS (SPARE6_STATUS):** Frequency is limited due to camera (device3) driver override, and specifically as a result of camera being power or thermal limited. IA cores are being forced to run at the camera driver requested level while camera is power or thermally performance limited.
5	0h RO/V	**DEV2_POWER_THERMAL_STATUS (SPARE5_STATUS):** Frequency is limited due to graphics (device2) driver override, and specifically as a result of graphics being power or thermal limited. IA cores are being forced to run at the graphics driver requested level while graphics is power or thermally performance limited.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RO/V	**POWER_BALANCER_STATUS (SPARE4_STATUS):** Frequency is limited due to cross-domain performance balancing as a result of package level power limits (either PL1 or PL2)
3	0h RO/V	**PL2_STATUS:** Frequency is limited due to a package-level RAPL PL2 excursion.
2	0h RO/V	**PL1_STATUS:** Frequency is limited due to a package-level RAPL PL1 excursion.
1	0h RO/V	**THERMAL_STATUS:** Frequency is limited due to thermal excursion.
0	0h RO/V	**PROCHOT_STATUS:** requency is limited due to external PROCHOT assertion, if supported on Broxton.

3.517 IA Core C0 Residency Counter (P_CR_TELEM_IA_C0_RESIDENCY_0_0_0_MCHBAR) — Offset 70C0h

This counter measures time that any core is active in the C0 state. This counter counts at the crystal clock frequency divided by 16. This counter may be used along with the IA Frequency Accumulator to calculate the average active clock ratio multiplier on the IA domain. **Average Active Frequency = Frequency Accumulator / C0 Residency**

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 70C0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**C0 Residency (DATA):** This counter measures time that any core is active in the C0 state. This counter counts at the crystal clock frequency divided by 16.

3.518 Graphics C0 Residency Counter (P_CR_TELEM_GT_C0_RESIDENCY_0_0_0_MCHBAR) — Offset 70C4h

This counter measures time that graphics is active in the C0 state. This counter counts at the crystal clock frequency divided by 16. This counter may be used along with the Graphics Frequency Accumulator to calculate the average active clock ratio multiplier on the GT domain. **Average Active Frequency = Frequency Accumulator / C0 Residency**

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 70C4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**C0 Residency (DATA):** This counter measures time that graphics is active in the C0 state. This counter counts at the crystal clock frequency divided by 16.

3.519 I-unit Processing System C0 Residency Counter (P_CR_TELEM_IUNIT_C0_RESIDENCY_0_0_0_MCHBAR) — Offset 70C8h

This counter measures time that I-unit processing system is active in the C0 state. This counter counts at the crystal clock frequency divided by 16. This counter may be used along with the I-unit Frequency Accumulator to calculate the average active clock ratio multiplier on the I-unit domain. **Average Active Frequency = Frequency Accumulator / C0 Residency**

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 70C8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**C0 Residency (DATA):** This counter measures time that I-unit processing system is active in the C0 state. This counter counts at the crystal clock frequency divided by 16.

3.520 TELEM_IA_FREQ_ACCUMULATOR (P_CR_TELEM_IA_FREQ_ACCUMULATOR_0_0_0_MCHBAR) — Offset 70CCh

Frequency accumulation data counted at ART >> 4

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 70CCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**DATA:** Residency data

3.521 Graphics C0 Residency Counter (P_CR_TELEM_GT_FREQ_ACCUMULATOR_0_0_0_MCHBAR) — Offset 70D0h

This counter integrates the current clock ratio multiplier for the Graphics domain at the same rate as the corresponding C0 residency counter. Its primary utility is in assessing the average active frequency of the domain **Average Active Frequency = Frequency Accumulator / C0 Residency**

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 70D0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Clock Ratio Multiplier Accumulator (DATA):** This counter integrates the current clock ratio of the domain at the same rate as the corresponding C0 residency counter

3.522 I-unit Processing System C0 Residency Counter (P_CR_TELEM_IUNIT_FREQ_ACCUMULATOR_0_0_0_MCHBAR) — Offset 70D4h

This counter integrates the current clock ratio multiplier for the I-unit processing system domain at the same rate as the corresponding C0 residency counter. Its primary utility is in assessing the average active frequency of the domain **Average Active Frequency = Frequency Accumulator / C0 Residency**

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 70D4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Clock Ratio Multiplier Accumulator (DATA):** This counter integrates the current clock ratio of the domain at the same rate as the corresponding C0 residency counter

3.523 Memory Active Residency (P_CR_TELEM_FAR_MEMORY_ACTIVE_0_0_0_MCHBAR) — Offset 70E8h

This counter measures the total time spent with memory active, as measured by any rank being in the active or active idle state. The inverse of this counter indicates the total time spent with all memory in the self-refresh state. This counter counts at the crystal clock frequency divided by 16.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 70E8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	0h RO/V	**DATA:** This counter measures the total time spent with memory active, as measured by any rank being in the active or active idle state. The inverse of this counter indicates the total time spent with all memory in the self-refresh state. This counter counts at the crystal clock frequency divided by 16.

3.524 Package Temperatures (P_CR_PACKAGE_TEMPERATURES_0_0_0_MCHBAR) — Offset 70F4h

Read-only register used for monitoring thermal status from all domains in the package.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 70F4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO/V	**System Agent Temperature (SA_TEMPERATURE):** System agent domain max temperature in degrees C. Reported in a signed, 2's complement format with the LSB representing 1'C resolution (S8.7.0). Raw, unfiltered
23:16	0h RO/V	**I-unit Temperature (ISP_TEMPERATURE):** Camera domain max temperature in degrees C. Reported in a signed, 2's complement format with the LSB representing 1'C resolution (S8.7.0). Raw, unfiltered
15:8	0h RO/V	**Graphics Temperature (GT_TEMPERATURE):** Graphics domain max temperature in degrees C. Reported in a signed, 2's complement format with the LSB representing 1'C resolution (S8.7.0). Raw, unfiltered
7:0	0h RO/V	**IA Core Temperature (IA_TEMPERATURE):** Virtual max temperature of all IA cores in degrees C. Reported in a signed, 2's complement format with the LSB representing 1'C resolution (S8.7.0). Raw, unfiltered

3.525 Package Thermal Limit Control (P_CR_THERMAL_LIMIT_CONTROL_0_0_0_MCHBAR) — Offset 7104h

This register is used for run-time control of the package level virtual thermal sensor. This interrupt and threshold is most commonly utilized by drivers wishing to control maximum silicon temperature in order to manage local or system level thermals due to various physical constraints. This temperature configuration applies only to in-die thermal sensors and not to any DRAM related thermal control.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7104h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Reserved (RSVD):** Reserved
8	0h RW	**ENABLE:** When set, it enables run-time thermal limit control to the THERMAL_LIMIT_TEMP described in this register.
7:0	0h RW	**Package Thermal Limit Temperature (THERMAL_LIMIT_TEMP):** Maximum SOC temperature allowed. Described in a signed, 2's complement format with the least significant bit representing 1'C resolution (S8.7.0). If the setting is higher than the processor's factory configured maximum temperature as described in the TEMPERATURE_TARGET MSR, this field is ignored. This field may be updated at any time.

3.526 Memory Subsystem Frequency Capabilities (P_CR_MEMSS_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR) — Offset 7108h

Describes the maximum frequency capabilities of DDR supported on this particular SOC. If the maximum supported frequency reports a zero, it indicates that the respective DRAM technology is not supported on this product.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7108h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RW	**Reserved (NUM_NM_CH):** Reserved
29:24	0h RW	**Reserved (WIO_FREQ):** Reserved
23:18	0h RW	**LPDDR4 Max Frequency (LP4_FREQ_HIGH):** This field indicates maximum LPDDR4 frequency that SOC supports, in integer multiple of 133.33MHz. A value of zero indicates LPDDR4 is not supported.
17:12	0h RW	**LPDDR4 Max Frequency at Min Voltage (LP4_FREQ_LOW):** This field indicates maximum LPDDR4 frequency supported at the minimum voltage level, in integer multiple of 133.33MHz. If this frequency is the same as 'Max' frequency, it indicates there is no voltage scaling. A value of zero indicates LPDDR4 is not supported.
11:6	0h RW	**LPDDR3 Max Frequency (LP3_FREQ_HIGH):** This field indicates maximum LPDDR3 frequency that SOC supports, in integer multiple of 133.33MHz. A value of zero indicates LPDDR3 is not supported.
5:0	0h RW	**LPDDR3 Max Frequency at Min Voltage (LP3_FREQ_LOW):** This field indicates maximum LPDDR3 frequency supported at the minimum voltage level, in integer multiple of 133.33MHz. If this frequency is the same as 'Max' frequency, it indicates there is no voltage scaling. A value of zero indicates LPDDR3 is not supported.

3.527 Memory Controller (MC) BIOS Reset Request and Status (P_CR_MC_BIOS_REQ_0_0_0_MCHBAR) — Offset 7114h

This register is used as the primary interface between BIOS and P-unit with respect to the Memory sub-system reset flow. This register provides both details about the current memory configuration as well as memory power-up sequencing controls. The typical memory subsystem reset and configuration flow is as follows. Each bullet describes the strict sequential ordering of the flow.

- BIOS reads MEMSS_FREQUENCY_CAPABILITIES to discover silicon capabilities.
- BIOS detects LPDDR DRAM frequency and the number of LPDDR channels that have DRAM devices attached. These results are then configured into the LPDDR channel active and frequency configuration fields.
- BIOS discovers if DRAM is currently in self-refresh, and if so, it configures Request Type (REQ_TYPE) to 100b to indicate this state. The SOC then uses this information as a way to ensure proper self-refresh exit flows as part of power-up sequencing.
- Once the lower 16 bits of the MC_BIOS_REQ register are configured based on platform discovery, BIOS sets Run/Busy to initiate firmware to start the memory subsystem reset sequence.

- P-unit firmware executes the initial configuration of PHY settings, and when complete it clears Run/Busy bit
- BIOS executes DDR PHY static configuration flow and initializes PHY PLL, and ensures that both steps are complete
- BIOS sets MC_BIOS_REQ.PHY_CONFIG_COMPLETE and MC_BIOS_REQ.RUN_BUSY to start the next phase
- P-unit firmware initiates the power-up sequence for D-units and Wide I/O collateral logic.
- P-unit firmware sets MC_BIOS_REQ.DUNIT_RESET_COMPLETE and clears the Run/Busy when it is done.
- BIOS observes Run/Busy deassertion and continues with MRC flow to initilize, train, configure memory subsystem settings.
- BIOS sets MC_BIOS_REQ.CPGC_MODE_COMPLETE and MC_BIOS_REQ.RUN_BUSY to start the next phase
- P-unit firmware configures D-unit, PHY to support normal operation of memory subsystem, and when completed, clears the Run/Busy bit
- BIOS waits for MC_BIOS_REQ.RUN_BUSY to clear, and then complete remaining configuration of Dunit and MLMC.
- BIOS executes memory configuration validation and lock sequence with the CSE.
- BIOS sets MC_BIOS_REQ.MEM_INIT_DONE and MC_BIOS_REQ.RUN_BUSY to indicate to Punit that all memory configuration is complete and memory configuration is locked.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7114h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Run/Busy (RUN_BUSY):** This bit indicates that the BIOS request is pending for P-unit firmware processing. BIOS sets this bit together with command details defined in the lower bits of this register. Firmware may only clear this bit after the BIOS request has been observed and completed. • 0 = The MC BIOS reset mailbox is idle or the last request has been completed. Software may initiate new requests. • 1 = The MC BIOS reset mailbox is busy. It is still handling a request. Writes to the mailbox are not allowed at this time.
30:27	0h RW	**RESERVED_2:** Reserved
26:24	0h RW	**DRAM_TYPE:** BIOS programs DRAM type filed: • 001b = Reserved • 010b = LPDDR4 • 100b = Reserved • 101b = DDR4 • else = reserved
23:20	0h RW	**RESERVED_1:** Reserved
19	0h RW	**Memory Init Done (MEM_INIT_DONE):** BIOS programs this bit after memory subsystem is fully configured, including security locking configuration completed

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW	**CPGC Mode Complete (CPGC_MODE_COMPLETE):** BIOS programs this bit after the memory train/init flow is complete. This initiates P-unit firmware execution of memory and D-unit clock configuration settings for normal operation
17	0h RW	**D-unit Reset Complete (DUNIT_RESET_COMPLETE):** Punit programs this bit after memory subsystem IPs are powered and corresponding reset flows are complete. At this point, those blocks are ready for executing the memory training flow, including initialization to support CPGC mode.
16	0h RW	**Memory PHY Configuration Complete (PHY_CONFIG_COMPLETE):** BIOS programs this bit indicating PHY initial configuration is complete and all DDR PHY PLLs are locked. Upon observation of this flag, P-unit firmware will initiate the power-up sequence of memory subsystem related IPs
15	0h RW	**DDR Channel 11 Active (FM_CH3_ACTIVE):** BIOS writes this value to indicate a DDR memory channel has DRAM devices active to allow the channel to have active memory traffic. Note, by default, a channel is not active and BIOS needs to explicitly program a value of 1 to indicate that the channel is active. If a channel is fused off on a particular SOC, BIOS input is ignored: • 0 = channel not active • 1 = channel active
14	0h RW	**DDR Channel 00 Active (FM_CH2_ACTIVE):** BIOS writes this value to indicate a DDR memory channel has DRAM devices active to allow the channel to have active memory traffic. Note, by default, a channel is not active and BIOS needs to explicitly program a value of 1 to indicate that the channel is active. If a channel is fused off on a particular SOC, BIOS input is ignored: • 0 = channel not active • 1 = channel active
13	0h RW	**DDR Channel 10 Active (FM_CH1_ACTIVE):** BIOS writes this value to indicate a DDR memory channel has DRAM devices active to allow the channel to have active memory traffic. Note, by default, a channel is not active and BIOS needs to explicitly program a value of 1 to indicate that the channel is active. If a channel is fused off on a particular SOC, BIOS input is ignored: • 0 = channel not active • 1 = channel active
12	0h RW	**DDR Channel 01 Active (FM_CH0_ACTIVE):** BIOS writes this value to indicate a DDR memory channel has DRAM devices active to allow the channel to have active memory traffic. Note, by default, a channel is not active and BIOS needs to explicitly program a value of 1 to indicate that the channel is active. If a channel is fused off on a particular SOC, BIOS input is ignored: • 0 = channel not active • 1 = channel active
11	0h RW	**RESERVED_2 (WIO_ONLY):** Reserved
10	0h RW	**RESERVED_0:** Reserved
9	0h RW	**RESERVED_3 (REQ_TYPE_SPECIAL):** Reserved
8:6	0h RW	**Request Type (REQ_TYPE):** This field is used to configure reset hints to the P-unit firmware. Encodings include: • 1xxb = Memory is in self-refresh, manual self-refresh exit is required • 0xxb = Memory is not in self-refresh or DRAM contents do not need to be preserved.
5:0	0h RW	**DDR Frequency Configuration (REQ_DATA):** BIOS programs this field to request DDR frequency in integer multiple of 133.33MHz. BIOS reads MEMSS_FREQUENCY_CAPABILITIES register(s) to discover maximum SOC supported capabilities. And BIOS is expected to request only legal DDR frequencies that are equal or lower than the maximum SOC supported capabilities.

3.528 P_CR_MEMSS_FREQUENCY_CAPABILITIES1_0_0_0_MCHBAR — Offset 7118h

Describes the maximum frequency capabilities of DDR supported and DDR configuration on this particular SOC. If the maximum supported frequency reports a zero, it indicates that the respective DRAM technology is not supported on this product.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7118h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RW	**RESERVED_2:** Reserved
24	0h RW	**DDR_CONFIG_LIMITATION:** Describe the DDR configuration limitation of this particular SOC: • 0 = no DDR configuration limitation • 1 = DDR configuration is limited to1ch x 64-bit DDR3L, or 2ch x32 LPDDR3 or LPDDR4
23:18	0h RW	**DDR4_FREQ_HIGH:** This field indicates maximum DDR4 frequency that SOC supports, in integer multiple of 133.33MHz. A value of zero indicates DDR4 is not supported
17:12	0h RW	**DDR4_FREQ_LOW:** This field indicates maximum DDR4 frequency supported at the minimum voltage level, in integer multiple of 133.33MHz. If this frequency is the same as Max frequency, it indicates there is no voltage scaling. A value of zero indicates DDR4 is not supported
11:6	0h RW	**DDR3L Max Frequency (DDR3L_FREQ_HIGH):** This field indicates maximum DDR3L frequency that SOC supports, in integer multiple of 133.33MHz. A value of zero indicates DDR3L is not supported.
5:0	0h RW	**DDR3L Max Frequency at Min Voltage (DDR3L_FREQ_LOW):** This field indicates maximum DDR3L frequency supported at the minimum voltage level, in integer multiple of 133.33MHz. If this frequency is the same as Max frequency, it indicates there is no voltage scaling. A value of zero indicates LPDDR4 is not supported.

3.529 PP1_C0_CORE_CLOCK_0_0_0_MCHBAR (P_CR_PP1_C0_CORE_CLOCK_0_0_0_MCHBAR) — Offset 7160h

GT RC0 residency counter. Holds the accumulated number of CS clks that GT has been in RC0.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7160h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**DATA:** Accumulated cycles GT has been in RC0.

3.530 Core Exists Vector (P_CR_CORE_EXISTS_VECTOR_0_0_0_MCHBAR) — Offset 7164h

Indication of the physical presence of IA cores in this silicon. IA core modules are defined as containing pairs of cores and an associated L2 cache. Module existance can therefore be inferred by OR'ing pairs of COREx_EXISTS in this register. This register does not reflect the impact of any software-based core disabling. It always reflects the capabilities of the silicon.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7164h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RSVD):** Reserved
3	0h RW	**CORE3_EXISTS:** Indication of core physical presence
2	0h RW	**CORE2_EXISTS:** Indication of core physical presence
1	0h RW	**CORE1_EXISTS:** Indication of core physical presence
0	0h RW	**CORE0_EXISTS:** Indication of core physical presence

3.531 Software Core Disable Mask (P_CR_CORE_DISABLE_MASK_0_0_0_MCHBAR) — Offset 7168h

Software may disable cores using this interface. The bit definition of this register exactly matches that defined in the CORE_EXISTS_VECTOR register. Punit firmware will apply the mask programmed into this register against the CORE_EXISTS_VECTOR to establish the resolved cores and modules to power up after a cold reset. The flow is as follows:

- Cold boot
- BIOS reads CORE_EXISTS_VECTOR to establish which modules and cores are present
- BIOS writes 1b to the corresponding core that it wishes to **disable**.
- BIOS may disable entire modules by writing 1b to a pair of cores.
- The results of the configuration are maintained in the sustain power well.
- BIOS initiates a cold reset flow at the platform. In a cold reset, the sustain power well maintains power most other rails are power cycled.
- On cold reset exit, Punit firmware inspects the core configuration and launches only the cores requested by BIOS. To software, it will appear as if these cores do not exist.

Software may discover the resolved core exists vector: resolved_core_exists_vector = (!CORE_DISABLE_MASK) & CORE_EXISTS_VECTOR

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7168h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RSVD):** Reserved
3	0h RW	**CORE3_DISABLE_MASK:** core3 disable mask
2	0h RW	**CORE2_DISABLE_MASK:** core2 disable mask
1	0h RW	**CORE1_DISABLE_MASK:** core1 disable mask
0	0h RW	**CORE0_DISABLE_MASK:** core0 disable mask

3.532 PL3 and PL4 Control (P_CR_PL3_CONTROL_0_0_0_MCHBAR) — Offset 71F0h

Control Power Limit 3 (PL3) and Power Limit 4 (PL4) using this register. This limit control is physically different from the same control in the IA core MSR space.

- PL3 is designed to clamp peak sustained power to levels supported by the battery or input power supply and as such manage lifetime degradation of that power delivery element. With PL3, peak power excursions above the limit are allowed so long as they do not exceed the configured duty cycle constraint in this register.
- PL4 is designed to clamp peak instantaneous power to levels below the max supported by the battery or input power supply. These clamps are implemented a priori and the SOC is guaranteed to constrain itself below the PL4 limit always.

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 71F0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW/L	**LOCK:** Write a 1b to lock this register until next reset. Once locked, no further updates may be written to any bits in the register.
62:48	0h RO	**Reserved (RSVD):** Reserved
47	0h RW/L	**PL4 Enable (PL4_ENABLE):** • 0 = disabled • 1 = enabled

Bit Range	Default & Access	Field Name (ID): Description
46:32	0h RW/L	**PL4 Max Power (PMAX):** Power Limit 'PL4' or Pmax power limit in the units as described PACAKGE_POWER_SKU_UNIT MSR. The SOC guarantees it will never exceed this power limit even for very short time windows.
31	0h RO	**Reserved (RSVD):** Reserved
30:24	0h RW/L	**PL3 Duty Cycle (DUTY_CYCLE):** Power limit excursion duty cycle control for PL3, describing what percentage of time it is allowed for the SOC to exceed the programmed PL3 power limit. 0% implies excursions are not supported ever and 100% implies excursions are always allowed (effectively disabling the feature). Units are in percentage(%). E.g., to allow for 20% excursion time and 80% PL3 power limit clamp time, program a value of 14h. Values greater than 100 (64h) are clipped to 100%.
23:17	0h RW/L	**PL3 Time Window (TIME_WINDOW):** Duration over which duty cycle control will be maintained. The bits of this field describe parameters for a mathematical equation for time window configuration. This time window is strictly adhered to, if the window described is 40ms, then silicon guarantees no excursions to the programmed duty cycle within a rolling 40ms window. This field is split into two sub-fields: • x = bits[6:5] • y = bits[4:0] Time window equation: **time_window = PACKAGE_POWER_SKU_UNIT.TIME_UNIT * ((1+x/4)^y)**
16	0h RO	**Reserved (RSVD):** Reserved
15	0h RW/L	**PL3 Enable (PL3_ENABLE):** • 0 = disabled • 1 = enabled
14:0	0h RW/L	**PL3 Power Limit (POWER_LIMIT):** Power Limit 3 (PL3) or PAppMax power level. Any SOC power measurement observed above this level is considered as an excursion against the PL3 power limit and duty cycle / time window budget. Units of this power limit are defined by PACKAGE_POWER_SKU_UNIT_MSR.PWR_UNIT.

3.533 Graphics Superqueue Active Clocks (P_CR_PP1_ANY_THREAD_ACTIVITY_0_0_0_MCHBAR) — Offset 7244h

Graphics Superqueue active residency counter. Counts at the crystal clock frequency divided by 16.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7244h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Superqueue Active Residency (SUPERQUEUE_ACTIVE_RESIDENCY):** Graphics Superqueue active residency counter. Counts in crystal reference clocks divided by 16.

3.534 LPDDR DRAM Thermal (MR4) Status of Channel 00 (P_CR_MEM_MR4_TEMPERATURE_DEV3_0_0_0_MCHBAR) — Offset 7248h

LPDDR DRAM Thermal (MR4) Status of Channel 00, when there are multiple DRAMs in a rank, the maximun MR4 is reported

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7248h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RO	**RESERVED_0:** Reserved
5:3	0h RW	**MR4 DRAM thermal status of Channel 00 Rank 1 (MR4_RANK_1):** This field is updated each read of LPDDR DRAM MR4 Device Temperature Status per rank. Update rate is configured by BIOS.
2:0	0h RW	**MR4 DRAM thermal status of Channel 00 Rank 0 (MR4_RANK_0):** This field is updated each read of LPDDR DRAM MR4 Device Temperature Status per rank. Update rate is configured by BIOS.

3.535 LPDDR DRAM Thermal (MR4) Status of Channel 11 (P_CR_MEM_MR4_TEMPERATURE_DEV4_0_0_0_MCHBAR) — Offset 724Ch

LPDDR DRAM Thermal (MR4) Status of Channel 11, when there are multiple DRAMs in a rank, the maximun MR4 is reported

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 724Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RO	**RESERVED_0:** Reserved
5:3	0h RW	**MR4 DRAM thermal status of Channel 11 Rank 1 (MR4_RANK_1):** This field is updated each read of LPDDR DRAM MR4 Device Temperature Status per rank. Update rate is configured by BIOS.
2:0	0h RW	**MR4 DRAM thermal status of Channel 11 Rank 0 (MR4_RANK_0):** This field is updated each read of LPDDR DRAM MR4 Device Temperature Status per rank. Update rate is configured by BIOS.

3.536 Interrupt Redirection Control (INTR_REDIR_CTL_MCHBAR) — Offset 7800h

Interrupt redirection control.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7800h	150000 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved 2 (RESERVED_2):** Reserved
21	0h RW	**INTPRIUP Gap DIS (INTPRIUP_GAP_DIS):** When the T-unit receives the data for intpriupdate IDI transaction, if the data specifies legacy clustered mode, then the T-Unit will assume there is a gap in the logical apic id field, such that: cluster_id[3:0] == logical_id[19:16] and agent_id[3:0] == logical_id[3:0]. These two fields are stored without a gap in the T-Unit CRs. If this CR bit is a one, then T-Unit will not assume a gap in the incoming data, and will just use the lower 8 bits of logical_id for the legacy cluster case.
20	1h RW	**INTPRIUP Enable (INTPRIUP_EN):** Enable an intpriupdate IDI transaction to modify the corresponding core's enable bit, physical APIC ID, and logical APIC ID.
19	0h RW	**INTPRIUP DFR CF (INTPRIUP_DFR_CF):** Enable an intpriupdate IDI transaction to modify the destination format bit (dt/dfr), which is common to all cores, if the core's APIC enable bit is set in the transaction data.
18	1h RW	**INTPRIUP DFR Enable (INTPRIUP_DFR_EN):** Enable an intpriupdate IDI transaction to modify the destination format bit (dt/dfr), which is common to all cores.
17	0h RW	**INTPRIUP X2APIC CF (INTPRIUP_X2APIC_CF):** Enable an intpriupdate IDI transaction to modify the extended APIC mode bit, which is common to all cores, if the core's APIC enable bit is set in the transaction data.
16	1h RW	**INTPRIUP X2APIC Enable (INTPRIUP_X2APIC_EN):** Enable an intpriupdate IDI transaction to modify the extended APIC mode bit, which is common to all cores.
15:9	0h RO	**Reserved 1 (RESERVED_1):** Reserved
8:6	0h RW	**Hash Vector (HASH_VECTOR):** Not supported. Vector-based interrupt redirection control: • 000: Select bits 7:4/5:4 for vector cluster/flat algorithm; • 001: Select bits 6:3/4:3; • 010: Select bits 4:1/2:1; • 011: Select bits 3:0/1:0; • Others: Reserved.
5:4	0h RO	**Reserved 0 (RESERVED_0):** Reserved
3	0h RW	**Cluster DESTFX DIS (CLUSTER_DESTFX_DIS):** When doing redirection on a logical cluster mode interrupt with a cluster id of all ones, the default is zero.
2:0	0h RW	**Redirection Mode Select (REDIR_MODE_SEL):** Redirection Mode Select for Logical Interrupts: • 000: Fixed priority • 001: Round robin • 010: Hash Vector Not Supported • 100: PAIR Power Aware Interrupt Redirection w/ Fixedpriority • 101: PAIR w/ Roundrobin Not Supported • 110: PAIR w/ Hash Vector Not Supported

3.537 X2B_BARB_CTL1 (X2B_BARB_CTL1_MCHBAR) — Offset 7804h

Various weights for the X2B data arbiter.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7804h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved 1 (RESERVED_1):** Reserved
23:22	0h RW	**IDI7 Write Data Weight (IDI7_WRITE_DATA_WEIGHT):** IDI Attach Point 7 Write Data Weight only exists if this IDI attach exists.
21:20	0h RW	**IDI6 Write Data Weight (IDI6_WRITE_DATA_WEIGHT):** IDI Attach Point 6 Write Data Weight only exists if this IDI attach exists.
19:18	0h RW	**IDI5 Write Data Weight (IDI5_WRITE_DATA_WEIGHT):** IDI Attach Point 5 Write Data Weight only exists if this IDI attach exists.
17:16	0h RW	**IDI4 Write Data Weight (IDI4_WRITE_DATA_WEIGHT):** IDI Attach Point 4 Write Data Weight only exists if this IDI attach exists.
15:14	0h RW	**IDI3 Write Data Weight (IDI3_WRITE_DATA_WEIGHT):** IDI Attach Point 3 Write Data Weight only exists if this IDI attach exists.
13:12	0h RW	**IDI2 Write Data Weight (IDI2_WRITE_DATA_WEIGHT):** IDI Attach Point 2 Write Data Weight only exists if this IDI attach exists.
11:10	0h RW	**IDI1 Write Data Weight (IDI1_WRITE_DATA_WEIGHT):** IDI Attach Point 1 Write Data Weight only exists if this IDI attach exists.
9:8	0h RW	**IDI0 Write Data Weight (IDI0_WRITE_DATA_WEIGHT):** IDI Attach Point 0 Write Data Weight only exists if this IDI attach exists.
7:2	0h RO	**Reserved 0 (RESERVED_0):** Reserved
1:0	0h RW	**PII2 Read Data Completion Weight (PII2_READ_DATA_COMP_WEIGHT):**

3.538 Clock Gating Control (CLKGATE_CTL_MCHBAR) — Offset 7808h

Each bit controls a separate Clock Gating Domain. BIOS should write all bits in this register to 1.

- 0: Overrides clock gating and forces the clock gating domain to behave like a freerunning clock.
- 1: Enables Clock Gating.

The reset value for this register is controlled by an SA strap. SOCs can tie this strap to a constant, a fuse or an output from another unit. NOTE: This has to be revisited for the new PND2 T-Unit structure clock gating control.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7808h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved 8 (RESERVED_8):** Reserved
30	0h RO	**Reserved 7 (RESERVED_7):** Reserved
29	0h RO	**Reserved 6 (RESERVED_6):** Reserved
28	0h RO	**Reserved 5 (RESERVED_5):** Reserved
27	0h RO	**Reserved 4 (RESERVED_4):** Reserved
26	0h RO	**Reserved 3 (RESERVED_3):** Reserved
25	0h RW	**IDI_U2C_CRDT_CNT_CLK_GATE_EN:** IDI U2C Credit Counters Clock Gate Enable
24	0h RW	**BT_FREE_CLK_GATE_EN:** BT Free Clock Gate Enable
23	0h RW	**Monitor Logic Clock Gate Enable (MON_LOG_CLK_GATE_EN):**
22	0h RW	**A2T Queue Clock Gate Enable (A2T_Q_CLK_GATE_EN):**
21	0h RW	**T2A Queue Clock Gate Enable (T2A_Q_CLK_GATE_EN):** T2A queue clock gate enable. NOTE: Need both request and writepull.
20	0h RW	**A2TAPIC Clock Gate Enable (A2TAPIC_CLK_GATE_EN):**
19	0h RW	**B2X Data Selector Clock Gate Enable (B2X_DATSEL_CLK_GATE_EN):**
18	0h RW	**X2B Data Selector Clock Gate Enable (X2B_DATSEL_CLK_GATE_EN):**
17	0h RW	**U2C Response Selector Clock Gate Enable (U2C_RESP_SEL_CLK_GATE_EN):**
16	0h RW	**T2A Request Selector Clock Gate Enable (T2A_REQ_SEL_CLK_GATE_EN):**
15	0h RW	**C2APIC FIFO Clock Gate Enable (C2APIC_FIFO_CLK_GATE_EN):**
14	0h RW	**U2C Request FIFO Clock Gate Enable (U2C_REQ_FIFO_CLK_GATE_EN):** NOTE: This should be the U2C request shim.
13	0h RW	**U2C Request Selector Clock Gate Enable (U2C_REQ_SEL_CLK_GATE_EN):** NOTE: Need per slice enable.
12	0h RW	**Upstream Ordering Block Clock Gate Enable (UOB_CLK_GATE_EN):**
11	0h RW	**Tracker Scoreboard Oldest Clock Gate Enable (TRKR_SB_OLDST_CLK_GATE_EN):** Tracker Scoreboard oldest of available queue clock gate enable.
10	0h RW	**Tracker Scoreboard U2C Response Status Clock Gate Enable (TRK_SCBD_U2C_RSP_STAT_CLK_GATE_EN):**
9	0h RW	**Tracker Scoreboard T2A Request Status Clock Gate Enable (TRKR_SB_T2A_REQSTAT_CLK_GATE_EN):**

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW	**Tracker Scoreboard B2X Data Status Clock Gate Enable (TRKR_SB_B2X_DATSTAT_CLK_GATE_EN):**
7	0h RW	**Tracker Scoreboard Write Status Clock Gate Enable (TRKR_SB_WRSTAT_CLK_GATE_EN):**
6	0h RW	**Tracker Scoreboard Snoop Status Clock Gate Enable (TRKR_SB_SNP_STAT_CLK_GATE_EN):**
5	0h RW	**Tracker Scoreboard Request Clock Gate Enable (TRKR_SB_REQ_CLK_GATE_EN):**
4	0h RW	**Tracker Scoreboard Violation Clock Gate Enable (TRKR_SB_VIOL_CLK_GATE_EN):**
3	0h RW	**Tracker Scoreboard Valid Clock Gate Enable (TRKR_SB_VALID_CLK_GATE_EN):**
2	0h RW	**Tracker Scoreboard Clock Gate Enable (TRKR_SB_CLK_GATE_EN):**
1	0h RW	**IOSF SB Config Register Clock Gate Enable (IOSF_SB_CFG_REG_CLK_GATE_EN):**
0	0h RW	**IOSF SB Message Clock Gate Enable (IOSF_SB_MSG_CLK_GATE_EN):**

3.539 Miscellaneous Controls (MISC_CTL_MCHBAR) — Offset 780Ch

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 780Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (RSVD):** Reserved
25:24	0h RO/V	**Snoop Ready Vector (SNP_RDY_VEC):** Allows Debug to dynamically read this state in the T-Unit.
23:18	0h RO	**Reserved (RSVD):** Reserved
17:16	0h RO/V	**Snoop Required Vector (SNP_REQ_VEC):** Allows Debug to dynamically read this state in the T-Unit.
15	0h RW	**C2U Data Tracking Disabled (C2U_DATA_TRACK_DIS):** When set, this bit disables the tracking of c2u_data for IDI shutdown qualification. A stop_idi sideband request requires that T-unit drain all u2c requests, and wait for all outstanding c2u responses and c2u data to return, before responding with the u2c_done sideband message to SoC.
14	0h RO	**IWB_FORWARDING_EN:** When an IWB comes into to B-unit, let it write the BRAM, even if the btag entry is locked waiting on the memory read return. Also, once written into the BRAM, let the tub2xdata logic consider the ooaq entry as ready to receive data. These two allowances will forward the IWB data to the requester, without waiting on the memory read return.
13:12	0h RW	**Spare 2 (SPARE2):** Spare bits for hardware.

Bit Range	Default & Access	Field Name (ID): Description
11:8	0h RW	**SPARE:** Readable and Writable Spare bits. SPARE[0] has been consumed. It is used by PCODE to bypass IDI_SHUTDOWN detection and will force IDI CSM FSM indicator to 1 for pgcb/idle detection.
7	0h RO	**Reserved (RSVD):** Reserved
6	0h RW	**GO-ACK FIFO Empty Check Disable (GO_ACK_FIFO_EMPTY_CHECK_DIS):** When set, the T-unit will NOT wait for the per-agent GO-ACK FIFO to be empty before accepting a stopidi request from the associated agent.
5	0h RW	**IDI_SHUTDOWN Bypass (IDI_SHUTDOWN_BYP):** Used by PCODE to bypass IDI_SHUTDOWN detection and will force IDI CSM FSM indicator to 1 for pgcb/idle detection.
4:1	0h RW	**DPTE Count (DPTE_CNT):** This field controls the number of cycles between sending Dynamic Prefetch Throttle Update Events to each IDI Attach Point. The 4-bit value in this field is N and the Count is 2 to the Nth power. • 0: T-Unit can send a DPTE every cycle. • 1: T-Unit can send a DPTE every 2 cycles. • 2: T-Unit can send a DPTE every 4 cycles. • 3: T-Unit can send a DPTE every 8 cycles. • 4: T-Unit can send a DPTE every 16 cycles. • 15: T-Unit can send a DPTE every 32768 cycles.
0	0h RW	**DPTE Enable (DPTE_EN):** • 0: Dynamic Prefetch Throttle Events are Disabled. • 1: Dynamic Prefetch Throttle Events are Enabled.

3.540 Control (CTL_MCHBAR) — Offset 7810h

Control register for various T-Unit knobs.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7810h	4B000000 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved 1 (RESERVED_1):** Reserved
30	1h RW	**ISOCH_BW_FIX_DIS:** Disable A-Unit blocking snooped traffic during IA/GT P-state transitions.
29:24	Bh RW	**ISOCH Shared Reserve (ISOCH_SHARED_RESV):** The number of shared T2A cmd credits to reserve for ISOCH level traffic.
23:22	0h RO	**Reserved 0 (RESERVED_0):** Reserved
21	0h RW	**Use PIC_End NACK (USE_PIC_END_NACK):** Normally, when GLM asserts pm_block_req and interrupt_ready is cleared, T-Unit finishes any active interrupt and then responds back with pic_end.
20	0h RW	**Disable Snooping GT (DISABLE_SNOOPING_GT):** • 0: Default. Snoop GT as normal. • 1: Never snoop GT for any access from an agent.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**Disable Opportunistic Slice 1 Schedule (DISABLE_OPPORTUNISTIC_SLICE1_SCHEDULE):** • 0: Default. Global B2X data selector will opportunistically schedule TSlice1 B2X selected data in the odd cycle if TSlice0 B2X selected data is not available. • 1: Global B2X data selector will always schedule TSlice1 B2X selected data only in the even cycle.
18	0h RW	**Disable IOSF High Priority Read Data Return (DISABLE_ISOC_HIGHPRI_RDDATA_RETURN):** Disables IOSF high priority read data returns and treats all read data scheduling as normal priority.
17	0h RW	**Enable NPC Collector (ENABLE_NPC_COLLECTOR):** This bit is not connected to anything.
16	0h RW	**Enable In Order APIC (ENABLE_IN_ORDER_APIC):** When 0, has no affect; When 1, forces all interrupts to bypass the piclets, and wait until all targeted cores are awake before delivering the interrupt. The APIC block will send int_wake to P-Unit for each targeted core. Note that using this bit could cause deadlock.
15	0h RW	**TG Enable High Priority Write Pulls (TG_HIGHPRI_WRITE_PULLS):** Enable High Priority Write Pulls. When 1 the X2B Data Selector will prioritize write pulls that the B-Unit has flagged as high priority over other write pulls.
14	0h RW	**TG Disable Live BRAM Bypass to PII2 (TG_LIVE_DATA_PII2):** Disable Live BRAM Bypass to PII2 Mode. When 1 all data to PII2 Agents will be returned from reading the BRAM.
13	0h RO	**TG Half Tunnel (TG_HALF_TUNNEL):** Enable IDI Half Tunneling. By Default IDI Half Tunneling is not required for proper functionality. This should only be enabled in an Intel Debug setting on Dual Core SoCs.
12	0h RW	**TG Disable nonDRAM Snoops (TG_NDRAMSNP):** Disable nonDRAM Snoops. • 0: SA will snoop nonDRAM Memory Addresses, including Memory Mapped IO MMIO and Memory Mapped Enhanced Config Space. • 1: SA will not snoop nonDRAM Memory Addresses. This includes the standard Memory Read/Write opcodes as well as SetMonitor. There is one exception since the SA will always issue Snoop Invalidate for CLFlush opcodes even if the address is not DRAM. Note: the Monitor Logic can still be armed and triggered using a nonDRAM address.
11	0h RO	**TG Read Data BW Mode (TG_RD_BW_MODE):** Read Data BW Mode. This Configuration Mode only affects behavior on SOCs that support two 64bit DDR3 Channels. For these SOCs this mode bit performs the following function: • 0: Enhanced Read BW mode for 2x64b DDR3 Channels. • 1: Basic Read BW mode. All other SOCs this mode bit performs the following function: • 0: Basic Read BW mode. • 1: Reserved.
10	0h RW	**TG Disable Downstream Posted Push Logic Mode (TG_DW_POST_PUSH_LOG):** Disable Downstream Posted Push Logic Mode. • 0: DRAM Read Completions to IOSF VC0 are stalled in the BRAM until all prior downstream Posted Requests from IDI are pushed into the Aunit. This ensures proper PCI Producer/Consumer Ordering Rules. • 1: Disables the Posted Push Logic. Note this mode violates the Producer/Consumer Ordering rules.
9	0h RW	**TG Set Monitor Snoop Configuration (TG_MON_SNP_CFG):** Set Monitor Snoop Configuration. Defines the Snoop opcode that will be sent for Set Monitor requests from IDI Agents: • 0:SnpCode default • 1:SnpInv

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW	**Broadcast APIC (BCAST_APIC):** Broadcast APIC Mode. When 1 T-Unit will wake up all IDI attach points and broadcast APIC messages to all attach points not just the IDI attach points that needs to receive the APIC message. Note: The T-Unit does not change the APIC ID to the broadcast APIC ID. What is meant by Broadcast is the Interrupt message is sent to all IDI Attach Points. For directed interrupts the directed interrupt is sent to all Attach Points with the unmodified APIC ID. For redirected interrupts the T-Unit performs the standard redirection algorithm fixed roundrobin or PAIR selects 1 APIC ID to send the message but then sends the message to all IDI Attach Points.
7	0h RW	**SNPINV:** SnpInv only when 1 T-Unit will only send SnpInv. This disables all other Snoop types e.g. SnpCode SnpData.
6	0h RW	**UC Code Read Snoop Configuration (UC_CODE_RD_CFG):** Code Read Snoop Configuration. The SA will send the following U2C Request for Code Reads from IDI Agents. • 0: SnpCode default. • 1: SnpInv.
5:4	0h RW	**PII2 Snooped Read (PII2_SNP_RD):** PII2 Snooped Read Configuration. The SA will send the following U2C Request for Snooped Reads from PII2 Agents. • 00b: Reserved. • 01b: SnpCode. • 10b: SnpData. • 11b: SnpInv default. Note: In order to use SnpCode or SnpData the B-Unit must be configured to disable the Owned Feature.
3	0h RW	**Always Snoop IDI Requests (ALWAYS_SNP_IDI):** IDI initiated requests will always be snooped even if Selfsnoop is not set and even if a Snoop Filter indicates no need to snoop.
2	0h RW	**Disable Live BRAM Bypass to IDI (DIS_LIVE_BRAM_BYP_IDI):** Disable Live BRAM Bypass to IDI Mode. When 1 all data to IDI Agents will be returned from reading the BRAM.
1	0h RW	**Outstanding Snoop (OUTSTND_SNP):** 1: Outstanding Snoop Mode. When 1, do not send any snoops if there is an outstanding Snoop from an IDI Read or Write to ECAM memory mapped Config space that has not yet received all Snoop Replies. Also, do not send any snoops if there is an outstanding Snoop from an IDI Read to MMIO that has not yet received all Snoop Replies.
0	0h RW	**Always Snoop PII2 Requests (ALWAYS_SNP_PII2):** Always Snoop PII2 Requests Mode. Before setting this bit note the following: There is a performance optimization in the B-Unit to allow multiple Snooped Writes from PII2 Agents to the same line. The B-Unit converts all but the 1st Snoop Write to be NonSnooped. Because of this, the Snoop Everything Mode should only be enabled if 1 of the following is true: T-Unit 1 Outstanding Snoop Mode is enabled OR B-Unit has been configured to disable request combining. Even when this mode is enabled the T-Unit will never snoop Memory Requests that map to MMIO from a PII2 Agent. These requests are treated as IMR violations and also the snoop is suppressed.

3.541 PSMI Control (PSMI_CTL_MCHBAR) — Offset 7814h

PSMI Control register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7814h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved 2 (RESERVED_2):** Reserved
7	0h RO	**Reserved 1 (RESERVED_1):** Reserved
6:5	0h RW	**Select IDI Attach Point (SEL_IDI_ATTACH):** Selects IDI Attach Point. i.e. Module to capture/replay. • 00b: IDI Attach Point 0 • 01b: IDI Attach Point 1 • 10b: Reserved • 11b: Reserved
4:3	0h RW	**PSMI Gear Ratio (PSMI_GEAR_RATIO):** PSMI Gear Ratio Select. • 00b: IDI transfers can occur every CZclock default • 01b: IDI transfers can occur every 2 CZclocks • 10b: IDI transfers can occur every 4 CZclocks • 11b: IDI transfers can occur every 8 CZclocks
2	0h RO	**Reserved (RESERVED_0):** Reserved
1	0h RW	**PSMI Replay Mode (PSMI_REPLAY_MODE):**
0	0h RW	**SA Capture Mode (SA_CAPTURE_MODE):** Note: If SA Capture Mode is enabled then SSA Replay Mode and P-Unit Replay Mode are ignored.

3.542 Miscellaneous T2A selector (T2A_SELECTOR_MISC_MCHBAR) — Offset 7818h

This register will provide miscellaneous features and controls for the T2A selector downstream block in the T-Unit.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7818h	380014A0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	38h RW	**PII_IDLE_THRESHOLD:** Number of 1x cycles that BT-unit PII idle conditions should exist before the ISM requests idle. The restart time of PII BGFs is on the order of 150 to 210 ns. To avoid this penalty, idle should not be entered too quickly.
23:15	0h RW	**Spare (SPARE1):** Spare register bits for read/write.
14:10	5h RW	**Maximum Downstream Nonposted Requests Issued For VC0b (MAX_DS_NP_VC0B):** Maximum number of downstream nonposted requests the T2A selector can issue for VC0b. This value should be programmed to less than or equal to the MAX_DS_NP field value but hardware will cap it at that value regardless.
9:5	5h RW	**Maximum Downstream Nonposted Requests Issued For VC0a (MAX_DS_NP_VC0A):** Maximum number of downstream nonposted requests the T2A selector can issue for VC0a. This value should be programmed to less than or equal to the MAX_DS_NP field value but hardware will cap it at that value regardless.
4:0	0h RW	**Spare (SPARE0):** Spare register bits for read/write.

3.543 VC Read Ordering CFG (VC_READ_ORDERING_CFG_MCHBAR) — Offset 781Ch

Register that indicates whether a particular B2T agent ID is configured to be inorder 1 or can be outoforder 0 for read accesses. This is will cause the access to be routed through the upstream ordering block value of 1 in the VCs bit or not value of 0 in the VCs bit. This register is not meant to be a dynamic ability to change inorder/outoforder nature and should be programmed by BIOS before devices are allowed to access memory. Note that reset value should work without a write. The IDI attaches are not upstream VCs or subject to the Upstream Ordering Block but to keep the handling generic bits will be hardwired to 0 for each IDI attach in the system.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 781Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved 0 (RESERVED_0):** Reserved
15	0h RW	**UPSTREAM_VC15 (UPSTREAM_VC15_IN_ORDER):** Configuration bit for upstream VC15 inorder read handling. 1 indicates that it is an inorder VC and reads must be handled by the Upstream Ordering Block. 0 indicates that it is an outoforder VC.
14	0h RW	**UPSTREAM_VC14 (UPSTREAM_VC14_IN_ORDER):** Configuration bit for upstream VC14 inorder read handling. 1 indicates that it is an inorder VC and reads must be handled by the Upstream Ordering Block. 0 indicates that it is an outoforder VC.
13	0h RW	**UPSTREAM_VC13 (UPSTREAM_VC13_IN_ORDER):** Configuration bit for upstream VC13 inorder read handling. 1 indicates that it is an inorder VC and reads must be handled by the Upstream Ordering Block. 0 indicates that it is an outoforder VC.
12	0h RW	**UPSTREAM_VC12 (UPSTREAM_VC12_IN_ORDER):** Configuration bit for upstream VC12 inorder read handling. 1 indicates that it is an inorder VC and reads must be handled by the Upstream Ordering Block. 0 indicates that it is an outoforder VC.
11	0h RW	**UPSTREAM_VC11 (UPSTREAM_VC11_IN_ORDER):** Configuration bit for upstream VC11 inorder read handling. 1 indicates that it is an inorder VC and reads must be handled by the Upstream Ordering Block. 0 indicates that it is an outoforder VC.
10	0h RW	**UPSTREAM_VC1BMMU_IN_ORDER:** Configuration bit for upstream VC1Bmmus inorder read handling. 1 indicates that it is an inorder VC and reads must be handled by the Upstream Ordering Block. 0 indicates that it is an outoforder VC.
9	0h RW	**UPSTREAM_VC0BMMU_IN_ORDER:** Configuration bit for upstream VC0Bmmus inorder read handling. 1 indicates that it is an inorder VC and reads must be handled by the Upstream Ordering Block. 0 indicates that it is an outoforder VC.
8	0h RW	**UPSTREAM_VC0AMMU_IN_ORDER:** Configuration bit for upstream VC0Ammus inorder read handling. 1 indicates that it is an inorder VC and reads must be handled by the Upstream Ordering Block. 0 indicates that it is an outoforder VC.
7	0h RW	**Upstream VCBR In Order (UPSTREAM_VCBR_IN_ORDER):** Configuration bit for upstream VCbrs inorder read handling. • 1: Indicates that it is an inorder VC and reads must be handled by the Upstream Ordering Block • 0: Indicates that it is an outoforder VC
6	0h RW	**Upstream VC2C In Order (UPSTREAM_VC2C_IN_ORDER):** Configuration bit for upstream VC2cs inorder read handling. • 1: Indicates that it is an inorder VC and reads must be handled by the Upstream Ordering Block • 0: Indicates that it is an outoforder VC

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**Upstream VC2B In Order (UPSTREAM_VC2B_IN_ORDER):** Configuration bit for upstream VC2bs inorder read handling. • 1: Indicates that it is an inorder VC and reads must be handled by the Upstream Ordering Block • 0: Indicates that it is an outoforder VC
4	0h RW	**Upstream VC2A In Order (UPSTREAM_VC2A_IN_ORDER):** Configuration bit for upstream VC2as inorder read handling. • 1 indicates that it is an inorder VC and reads must be handled by the Upstream Ordering Block. • 0 indicates that it is an outoforder VC.
3	0h RW	**Upstream VC1B In Order (UPSTREAM_VC1B_IN_ORDER):** Configuration bit for upstream VC1bs inorder read handling. • 1 indicates that it is an inorder VC and reads must be handled by the Upstream Ordering Block. • 0 indicates that it is an outoforder VC.
2	0h RW	**Upstream VC1A In Order (UPSTREAM_VC1A_IN_ORDER):** Configuration bit for upstream VC1as inorder read handling. • 1 indicates that it is an inorder VC and reads must be handled by the Upstream Ordering Block. • 0 indicates that it is an outoforder VC.
1	0h RW	**Upstream VC0B In Order (UPSTREAM_VC0B_IN_ORDER):** Configuration bit for upstream VC0bs inorder read handling. • 1 indicates that it is an inorder VC and reads must be handled by the Upstream Ordering Block. • 0 indicates that it is an outoforder VC.
0	0h RW	**Upstream VC0A In Order (UPSTREAM_VC0A_IN_ORDER):** Configuration bit for upstream VC0as inorder read handling. • 1 indicates that it is an inorder VC and reads must be handled by the Upstream Ordering Block. • 0 indicates that it is an outoforder VC.

3.544 VC Write Ordering CFG (VC_WRITE_ORDERING_CFG_MCHBAR) — Offset 7820h

Register that indicates whether a particular B2T agent ID is configured to be inorder 1 or can be outoforder 0 for write accesses. This is will cause the access to be routed through the upstream ordering block value of 1 in the VCs bit or not value of 0 in the VCs bit. This register is not meant to be a dynamic ability to change inorder/outoforder nature and should be programmed by BIOS before devices are allowed to access memory note that reset value should work without a write. The IDI attaches are not upstream VCs or subject to the Upstream Ordering Block but to keep the handling generic bits will be hardwired to 0 for each IDI attach in the system.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7820h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved 0 (RESERVED_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**UPSTREAM_VC15 (UPSTREAM_VC15_IN_ORDER):** Configuration bit for upstream VC15 inorder write handling. 1 indicates that it is an inorder VC and writes must be handled by the Upstream Ordering Block. 0 indicates that it is an outoforder VC.
14	0h RW	**UPSTREAM_VC14 (UPSTREAM_VC14_IN_ORDER):** Configuration bit for upstream VC14 inorder write handling. 1 indicates that it is an inorder VC and writes must be handled by the Upstream Ordering Block. 0 indicates that it is an outoforder VC.
13	0h RW	**UPSTREAM_VC13 (UPSTREAM_VC13_IN_ORDER):** Configuration bit for upstream VC13 inorder write handling. 1 indicates that it is an inorder VC and writes must be handled by the Upstream Ordering Block. 0 indicates that it is an outoforder VC.
12	0h RW	**UPSTREAM_VC12 (UPSTREAM_VC12_IN_ORDER):** Configuration bit for upstream VC12 inorder write handling. 1 indicates that it is an inorder VC and writes must be handled by the Upstream Ordering Block. 0 indicates that it is an outoforder VC.
11	0h RW	**UPSTREAM_VC11 (UPSTREAM_VC11_IN_ORDER):** Configuration bit for upstream VC11 inorder write handling. 1 indicates that it is an inorder VC and writes must be handled by the Upstream Ordering Block. 0 indicates that it is an outoforder VC.
10	0h RW	**UPSTREAM_VC1BMMU_IN_ORDER:** Configuration bit for upstream VC1Bmmus inorder write handling. 1 indicates that it is an inorder VC and writes must be handled by the Upstream Ordering Block. 0 indicates that it is an outoforder VC.
9	0h RW	**UPSTREAM_VC0BMMU_IN_ORDER:** Configuration bit for upstream VC0Bmmus inorder write handling. 1 indicates that it is an inorder VC and writes must be handled by the Upstream Ordering Block. 0 indicates that it is an outoforder VC.
8	0h RW	**Upstream VC0MMU In Order (UPSTREAM_VC0AMMU_IN_ORDER):** Configuration bit for upstream VC0mmus inorder write handling. • 1 indicates that it is an inorder VC and writes must be handled by the Upstream Ordering Block. • 0 indicates that it is an outoforder VC.
7	0h RW	**Upstream VCBR In Order (UPSTREAM_VCBR_IN_ORDER):** Configuration bit for upstream VCbrs inorder write handling. • 1 indicates that it is an inorder VC and writes must be handled by the Upstream Ordering Block. • 0 indicates that it is an outoforder VC.
6	0h RW	**Upstream VC2C In Order (UPSTREAM_VC2C_IN_ORDER):** Configuration bit for upstream VC2cs inorder write handling. • 1 indicates that it is an inorder VC and writes must be handled by the Upstream Ordering Block. • 0 indicates that it is an outoforder VC.
5	0h RW	**Upstream VC2B In Order (UPSTREAM_VC2B_IN_ORDER):** Configuration bit for upstream VC2bs inorder write handling. • 1 indicates that it is an inorder VC and writes must be handled by the Upstream Ordering Block. • 0 indicates that it is an outoforder VC.
4	0h RW	**Upstream VC2A In Order (UPSTREAM_VC2A_IN_ORDER):** Configuration bit for upstream VC2as inorder write handling. • 1 indicates that it is an inorder VC and writes must be handled by the Upstream Ordering Block. • 0 indicates that it is an outoforder VC.
3	0h RW	**Upstream VC1B In Order (UPSTREAM_VC1B_IN_ORDER):** Configuration bit for upstream VC1bs inorder write handling. • 1 indicates that it is an inorder VC and writes must be handled by the Upstream Ordering Block. • 0 indicates that it is an outoforder VC.
2	0h RW	**Upstream VC1A In Order (UPSTREAM_VC1A_IN_ORDER):** Configuration bit for upstream VC1as inorder write handling. • 1 indicates that it is an inorder VC and writes must be handled by the Upstream Ordering Block. • 0 indicates that it is an outoforder VC.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Upstream VC0B In Order (UPSTREAM_VC0B_IN_ORDER):** Configuration bit for upstream VC0bs inorder write handling. • 1 indicates that it is an inorder VC and writes must be handled by the Upstream Ordering Block. • 0 indicates that it is an outoforder VC.
0	0h RW	**Upstream VC0A In Order (UPSTREAM_VC0A_IN_ORDER):** Configuration bit for upstream VC0as inorder write handling. • 1 indicates that it is an inorder VC and writes must be handled by the Upstream Ordering Block. • 0 indicates that it is an outoforder VC.

3.545 IDI0 C2U Credit Control (IDI0_C2U_CREDIT_CTRL_MCHBAR) — Offset 7824h

This register will provide the number of credits that IDI C2U channels should be initialized to and other control mechanisms around IDI start/stop flows. A register will be provided per IDI attach to allow for individual IDI tuning. This register will be for the agent attached to IDI0.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7824h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Reserved 0 (RESERVED_0):** Reserved
17:12	0h RW	**IDI IA C2U Data Credit Initialization (IDI_IA_C2U_DATA_CREDIT_INIT):** This register is the number of credits to initialize C2U data credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U ingress data queue size in B-Unit. This provides an ability to limit data credits at the next IDI start request change will only take place after the next IDI start request. The default value of this register should work out of reset.
11:6	0h RW	**IDI IA C2U Response Credit Initialization (IDI_IA_C2U_RSP_CREDIT_INIT):** This register is the number of credits to initialize C2U response credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U response shim ingress queue size in T-Unit. This provides an ability to limit response credits at the next IDI start request. Change will only take place after next IDI start request. The default value of this register should work out of reset.
5:0	0h RW	**IDI IA C2U Request Credit Initialization (IDI_IA_C2U_REQ_CREDIT_INIT):** This register is the number of credits to initialize C2U requests credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U ingress queue size in Badmit. This provides an ability to limit request credits at the next IDI start request. Change will only take place after next IDI start request. The default value of this register should work out of reset.

3.546 IDI1 C2U Credit Control (IDI1_C2U_CREDIT_CTRL_MCHBAR) — Offset 7828h

This register will provide the number of credits that IDI C2U channels should be initialized to and other control mechanisms around IDI start/stop flows. A register will be provided per IDI attach to allow for individual IDI tuning. This register will be for the agent attached to IDI1.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7828h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Reserved 0 (RESERVED_0):** Reserved
17:12	0h RW	**IDI IA C2U Data Credit Initialization (IDI_IA_C2U_DATA_CREDIT_INIT):** This register is the number of credits to initialize C2U data credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U ingress data queue size in B-Unit. This provides an ability to limit data credits at the next IDI start request. Change will only take place after next IDI start request. The default value of this register should work out of reset.
11:6	0h RW	**IDI IA C2U Response Credit Initialization (IDI_IA_C2U_RSP_CREDIT_INIT):** This bit field contains the number of credits to initialize C2U response credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U response shim ingress queue size in T-Unit. This provides an ability to limit response credits at the next IDI start request. Change will only take place after next IDI start request. The default value of this register should work out of reset.
5:0	0h RW	**IDI IA C2U Request Credit Initialization (IDI_IA_C2U_REQ_CREDIT_INIT):** This register is the number of credits to initialize C2U requests credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U ingress queue size in Badmit. This provides an ability to limit request credits at the next IDI start request change will only take place after next IDI start request. The default value of this register should work out of reset.

3.547 IDI2 C2U Credit Control (IDI2_C2U_CREDIT_CTRL_MCHBAR) — Offset 782Ch

This register will provide the number of credits that IDI C2U channels should be initialized to and other control mechanisms around IDI start / stop flows. A register will be provided per IDI attach to allow for individual IDI tuning. This register will be for the agent attached to IDI2.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 782Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Reserved 0 (RESERVED_0):** Reserved
17:12	0h RO	**IDI C2U Data Credit Initialization (IDI_C2U_DATA_CREDIT_INIT):** This register is the number of credits to initialize C2U data credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U ingress data queue size in B-Unit. This provides an ability to limit data credits at the next IDI start request change will only take place after next IDI start request. The default value of this register should work out of reset.

Bit Range	Default & Access	Field Name (ID): Description
11:6	0h RO	**IDI C2U Response Credit Initialization (IDI_C2U_RSP_CREDIT_INIT):** This register is the number of credits to initialize C2U response credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U response shim ingress queue size in T-Unit. This provides an ability to limit response credits at the next IDI start request change will only take place after next IDI start request. The default value of this register should work out of reset.
5:0	0h RO	**IDI C2U Request Credit Initialization (IDI_C2U_REQ_CREDIT_INIT):** This register is the number of credits to initialize C2U requests credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U ingress queue size in Badmit. This provides an ability to limit request credits at the next IDI start request change will only take place after next IDI start request. The default value of this register should work out of reset.

3.548 IDI3 C2U Credit Control (IDI3_C2U_CREDIT_CTRL_MCHBAR) — Offset 7830h

This register will provide the number of credits that IDI C2U channels should be initialized to and other control mechanisms around IDI start/stop flows. A register will be provided per IDI attach to allow for individual IDI tuning. This register will be for the agent attached to IDI3.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7830h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Reserved 0 (RESERVED_0):** Reserved
17:12	0h RO	**IDI IA C2U Data Credit Initialization (IDI_IA_C2U_DATA_CREDIT_INIT):** This register is the number of credits to initialize C2U data credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U ingress data queue size in B-Unit. This provides an ability to limit data credits at the next IDI start request change will only take place after next IDI start request. The default value of this register should work out of reset.
11:6	0h RO	**IDI IA C2U Response Credit Initialization (IDI_IA_C2U_RSP_CREDIT_INIT):** This register is the number of credits to initialize C2U response credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U response shim ingress queue size in T-Unit. This provides an ability to limit response credits at the next IDI start request change will only take place after next IDI start request. The default value of this register should work out of reset.
5:0	0h RO	**IDI IA C2U Request Credit Initialization (IDI_IA_C2U_REQ_CREDIT_INIT):** This register is the number of credits to initialize C2U requests credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U ingress queue size in Badmit. This provides an ability to limit request credits at the next IDI start request change will only take place after next IDI start request. The default value of this register should work out of reset.

3.549 IDI4 C2U Credit Control (IDI4_C2U_CREDIT_CTRL_MCHBAR) — Offset 7834h

This register will provide the number of credits that IDI C2U channels should be initialized to and other control mechanisms around IDI start/stop flows. A register will be provided per IDI attach to allow for individual IDI tuning. This register will be for the agent attached to IDI4.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7834h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Reserved 0 (RESERVED_0):** Reserved
17:12	0h RO	**IDI IA C2U Data Credit Initialization (IDI_IA_C2U_DATA_CREDIT_INIT):** This register is the number of credits to initialize C2U data credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U ingress data queue size in B-Unit. This provides an ability to limit data credits at the next IDI start request change will only take place after next IDI start request. The default value of this register should work out of reset.
11:6	0h RO	**IDI IA C2U Response Credit Initialization (IDI_IA_C2U_RSP_CREDIT_INIT):** This register is the number of credits to initialize C2U response credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U response shim ingress queue size in T-Unit. This provides an ability to limit response credits at the next IDI start request change will only take place after next IDI start request. The default value of this register should work out of reset.
5:0	0h RO	**IDI IA C2U Request Credit Initialization (IDI_IA_C2U_REQ_CREDIT_INIT):** This register is the number of credits to initialize C2U requests credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U ingress queue size in Badmit. This provides an ability to limit request credits at the next IDI start request change will only take place after next IDI start request. The default value of this register should work out of reset.

3.550 IDI5 C2U Credit Control (IDI5_C2U_CREDIT_CTRL_MCHBAR) — Offset 7838h

This register will provide the number of credits that IDI C2U channels should be initialized to and other control mechanisms around IDI start/stop flows. A register will be provided per IDI attach to allow for individual IDI tuning. This register will be for the agent attached to IDI5.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7838h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Reserved 0 (RESERVED_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
17:12	0h RO	**IDI IA C2U Data Credit Initialization (IDI_IA_C2U_DATA_CREDIT_INIT):** This register is the number of credits to initialize C2U data credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U ingress data queue size in B-Unit. This provides an ability to limit data credits at the next IDI start request change will only take place after next IDI start request. The default value of this register should work out of reset.
11:6	0h RO	**IDI IA C2U Response Credit Initialization (IDI_IA_C2U_RSP_CREDIT_INIT):** This register is the number of credits to initialize C2U response credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U response shim ingress queue size in T-Unit. This provides an ability to limit response credits at the next IDI start request change will only take place after next IDI start request. The default value of this register should work out of reset.
5:0	0h RO	**IDI IA C2U Request Credit Initialization (IDI_IA_C2U_REQ_CREDIT_INIT):** This register is the number of credits to initialize C2U requests credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U ingress queue size in Badmit. This provides an ability to limit request credits at the next IDI start request change will only take place after next IDI start request. The default value of this register should work out of reset.

3.551 IDI6 C2U Credit Control (IDI6_C2U_CREDIT_CTRL_MCHBAR) — Offset 783Ch

This register will provide the number of credits that IDI C2U channels should be initialized to and other control mechanisms around IDI start/stop flows. A register will be provided per IDI attach to allow for individual IDI tuning. This register will be for the agent attached to IDI6.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 783Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Reserved 0 (RESERVED_0):** Reserved
17:12	0h RO	**IDI IA C2U Data Credit Initialization (IDI_IA_C2U_DATA_CREDIT_INIT):** This register is the number of credits to initialize C2U data credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U ingress data queue size in B-Unit. This provides an ability to limit data credits at the next IDI start request change will only take place after next IDI start request. The default value of this register should work out of reset.
11:6	0h RO	**IDI IA C2U Response Credit Initialization (IDI_IA_C2U_RSP_CREDIT_INIT):** This register is the number of credits to initialize C2U response credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U response shim ingress queue size in T-Unit. This provides an ability to limit response credits at the next IDI start request change will only take place after next IDI start request. The default value of this register should work out of reset.
5:0	0h RO	**IDI IA C2U Request Credit Initialization (IDI_IA_C2U_REQ_CREDIT_INIT):** This register is the number of credits to initialize C2U requests credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U ingress queue size in Badmit. This provides an ability to limit request credits at the next IDI start request change will only take place after next IDI start request. The default value of this register should work out of reset.

3.552 IDI7 C2U Credit Control (IDI7_C2U_CREDIT_CTRL_MCHBAR) — Offset 7840h

This register will provide the number of credits that IDI C2U channels should be initialized to and other control mechanisms around IDI start/stop flows. A register will be provided per IDI attach to allow for individual IDI tuning. This register will be for the agent attached to IDI4.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7840h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Reserved 0 (RESERVED_0):** Reserved
17:12	0h RO	**IDI IA C2U Data Credit Initialization (IDI_IA_C2U_DATA_CREDIT_INIT):** This register is the number of credits to initialize C2U data credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U ingress data queue size in B-Unit. This provides an ability to limit data credits at the next IDI start request change will only take place after next IDI start request. The default value of this register should work out of reset.
11:6	0h RO	**IDI IA C2U Response Credit Initialization (IDI_IA_C2U_RSP_CREDIT_INIT):** This register is the number of credits to initialize C2U response credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U response shim ingress queue size in T-Unit. This provides an ability to limit response credits at the next IDI start request change will only take place after next IDI start request. The default value of this register should work out of reset.
5:0	0h RO	**IDI IA C2U Request Credit Initialization (IDI_IA_C2U_REQ_CREDIT_INIT):** This register is the number of credits to initialize C2U requests credits for the IDI attach point associated with this register. It is not legal to program this field to a value larger than the C2U ingress queue size in Badmit. This provides an ability to limit request credits at the next IDI start request change will only take place after next IDI start request. The default value of this register should work out of reset.

3.553 PII2 A2T Credit Control (PII2_A2T_CREDIT_CTRL_MCHBAR) — Offset 7844h

This register will provide the number of credits that the T-Unit will initialize for the A2T credit initialization on PII2.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:0, F:0] MCHBAR + 7844h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved 0 (RESERVED_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RW	**PII2 A2T VC0B Credit Initialization (PII2_A2T_VC0B_CREDIT_INIT):** This field is the number of credits to initialize A2T VC0b requests credits. It is not legal to program this field to a value larger than the A2T VC0b FIFO ingress queue size in the UAM block. This provides an ability to limit request credits at the PII2 credit initialization change will only take place at next credit initialization. The default value of this register should work out of reset.
7:0	0h RW	**PII2 A2T VC0A Credit Initialization (PII2_A2T_VC0A_CREDIT_INIT):** This field is the number of credits to initialize A2T VC0a requests credits. It is not legal to program this field to a value larger than the A2T VC0a FIFO ingress queue size in the UAM block. This provides an ability to limit request credits at the PII2 credit initialization change will only take place at next credit initialization. The default value of this register should work out of reset.

3.554 BIOSWR Control Policy (T_CR_BIOSWR_CP_0_0_0_MCHBAR) — Offset 7848h

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 7848h	C0061010202 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	C0061010202h RW	**Access Control Policy (ACCESS_CTRL_POL):** The Access Control Policy for this register and the T-Unit Read/write Policy Registers.

3.555 BIOSWR Read Access Control (T_CR_BIOSWR_RAC_0_0_0_MCHBAR) — Offset 7850h

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 7850h	80000C00630D0217 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	80000C00630D0217h RW	**Read Policy (READ_POL):** The Read Policy for the T-Unit Registers.

3.556 BIOSWR Write Access Control (T_CR_BIOSWR_WAC_0_0_0_MCHBAR) — Offset 7858h

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 7858h	C00610C0212 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	C00610C0212h RW	**Write Policy (WRITE_POL):** The Write Policy for the T-Unit Registers.

3.557 Tunit Pcode/Ucode Write, All Read Control Policy Register (T_CR_P_U_CODEWR_ALLRD_CP_0_0_0_MCHBAR) — Offset 7860h

TUnit Pcode/Ucode Write, All Read control policy: This register controls the access policy to the Tunit P_U_CODEWR_ALLRD_RAC P_U_CODEWR_ALLRD_WAC

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 7860h	400010002 0 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	400010002 0h RW	**ACCESS_CTRL_POL:** The Access Control Policy for this register and the Tunit ucode Read/write Policy Registers.

3.558 TUnit Pcode/Ucode Read Access Control (T_CR_P_U_CODEWR_ALLRD_RAC_0_0_0_MCHBAR) — Offset 7868h

Pcode/Ucode Write, All Read Access Control Policy: This register controls the read access policy to the Tunit P_U_CODEWR_ALLRD

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 7868h	FFFFFFFFFFFFFFFF h

Bit Range	Default & Access	Field Name (ID): Description
63:0	FFFFFFFFFFFFFFFFh RO	**READ_POL:** The Read Policy for the Tunit ucode Registers.

3.559 TUnit Pcode/Ucode Write Access Control (T_CR_P_U_CODEWR_ALLRD_WAC_0_0_0_MCHBAR) — Offset 7870h

Pcode/Ucode Write, All Read Write Access Control Policy: This register controls the write access policy to the Tunit P_U_CODEWR_ALLRD

Type	Size	Offset	Default
MEM	64 bit	[B:0, D:0, F:0] MCHBAR + 7870h	40801008602 h

Bit Range	Default & Access	Field Name (ID): Description
63:0	40801008602h RW	**WRITE_POL:** The Write Policy for the Tunit ucode Registers.

3.560 PMTA_WEIGHTED_ARB_CONFIG (PMTA_WEIGHTED_ARB_CONFIG)—Offset 4C04h

PMTA_WEIGHTED_ARB_CONFIG

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 12204011h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**RSVD (RSVD):** Reserved
29	0h RW	**CFG_INITIAL_LINK_PREFERENCE_COMPQM (CFG_INITIAL_LINK_PREFERENCE_COMPQM):** Initial link preference for MEE completion queue. Preference is used to break ties between contending requests with same urgency value.
28	1h RW	**CFG_INITIAL_LINK_PREFERENCE_COMPQB (CFG_INITIAL_LINK_PREFERENCE_COMPQB):** Initial link preference for non-MEE completion queue. Preference is used to break ties between contending requests with same urgency value.

Bit Range	Default & Access	Field Name (ID): Description
27:24	2h RW	**CFG_INITIAL_LINK_WEIGHT_COMPQM (CFG_INITIAL_LINK_WEIGHT_COMPQM):** Initial link weight for MEE completion queue. Once the queue wins arbitration, it will stay as the winner for as many cycles as the weight assigned, provided there are responses available in the queue.
23:20	2h RW	**CFG_INITIAL_LINK_WEIGHT_COMPQB (CFG_INITIAL_LINK_WEIGHT_COMPQB):** Initial link weight for non-MEE completion queue. Once the queue wins arbitration, it will stay as the winner for as many cycles as the weight assigned, provided there are responses available in the queue.
19:16	0h RO	Reserved.
15:14	1h RW	**CFG_INITIAL_LINK_PREFERENCE_REQQM (CFG_INITIAL_LINK_PREFERENCE_REQQM):** Initial link preference for MEE request queue. Preference is used to break ties between contending requests with same urgency value.
13:12	0h RW	**CFG_INITIAL_LINK_PREFERENCE_REQQB (CFG_INITIAL_LINK_PREFERENCE_REQQB):** Initial link preference for non-MEE request queue. Preference is used to break ties between contending requests with same urgency value.
11:8	0h RO	Reserved.
7:4	1h RW	**CFG_INITIAL_LINK_WEIGHT_REQQM (CFG_INITIAL_LINK_WEIGHT_REQQM):** Initial link weight for MEE request queue. Once the queue wins arbitration, it will stay as the winner for as many cycles as the weight assigned, provided there are requests available in the queue.
3:0	1h RW	**CFG_INITIAL_LINK_WEIGHT_REQQB (CFG_INITIAL_LINK_WEIGHT_REQQB):** Initial link weight for non-MEE request queue. Once the queue wins arbitration, it will stay as the winner for as many cycles as the weight assigned, provided there are requests available in the queue.

3.561 PMTA_QOS_CONFIG0 (PMTA_QOS_CONFIG0)—Offset 4C08h

This register defines the urgency values for each of the access classes.

Access Method

Type: MEM Register (Size: 64 bits)

Device:
Function:

Default: 808000408000h

Bit Range	Default & Access	Field Name (ID): Description
63:56	0h RO	**RSVD (RSVD):** cfg_impending_isoc_req_urg_incr input to REQQX
55:48	0h RW	**URG6_INITIAL_URGENCY (URG6_INITIAL_URGENCY):** Initial urgency for urgency class 6
47:40	80h RW	**URG5_INITIAL_URGENCY (URG5_INITIAL_URGENCY):** Initial urgency for urgency class 5
39:32	80h RW	**URG4_INITIAL_URGENCY (URG4_INITIAL_URGENCY):** Initial urgency for urgency class 4
31:24	0h RW	**URG3_INITIAL_URGENCY (URG3_INITIAL_URGENCY):** Initial urgency for urgency class 3
23:16	40h RW	**URG2_INITIAL_URGENCY (URG2_INITIAL_URGENCY):** Initial urgency for urgency class 2
15:8	80h RW	**URG1_INITIAL_URGENCY (URG1_INITIAL_URGENCY):** Initial urgency for urgency class 1
7:0	0h RW	**URG0_INITIAL_URGENCY (URG0_INITIAL_URGENCY):** Initial urgency for urgency class 0

3.562 PMTA_QOS_CONFIG1 (PMTA_QOS_CONFIG1)—Offset 4C10h

This register defines the urgency delta values for each of the urgency classes. It also carries the threshold values for classifying AC1 requests to the right urgency class

Access Method

Type: MEM Register (Size: 64 bits)

Device:
Function:

Default: 40001484442h

Bit Range	Default & Access	Field Name (ID): Description
63:48	0h RO	**RSVD (RSVD):** Reserved
47:38	10h RW	**AC1_THR1 (AC1_THR1):** Deadline threshold for AC1 requests. All AC1 requests with deadline less than AC1_THR1 in the future are classified as urgency class 2 requests
37:28	0h RW	**AC1_THR0 (AC1_THR0):** Deadline threshold for AC1 requests. All AC1 requests with deadline less than AC1_THR0 in the future are classified as urgency class 1 requests. AC1 requests not mapping to urgency class 1 or 2 are classified as urgency class 3 requests
27:24	1h RW	**URG6_URGENCY_DELTA (URG6_URGENCY_DELTA):** Urgency delta for urgency class 6

Bit Range	Default & Access	Field Name (ID): Description
23:20	4h RW	**URG5_URGENCY_DELTA (URG5_URGENCY_DELTA):** Urgency delta for urgency class 5
19:16	8h RW	**URG4_URGENCY_DELTA (URG4_URGENCY_DELTA):** Urgency delta for urgency class 4
15:12	4h RW	**URG3_URGENCY_DELTA (URG3_URGENCY_DELTA):** Urgency delta for urgency class 3
11:8	4h RW	**URG2_URGENCY_DELTA (URG2_URGENCY_DELTA):** Urgency delta for urgency class 2
7:4	4h RW	**URG1_URGENCY_DELTA (URG1_URGENCY_DELTA):** Urgency delta for urgency class 1
3:0	2h RW	**URG0_URGENCY_DELTA (URG0_URGENCY_DELTA):** Urgency delta for urgency class 0

3.563 PMTA_MEE_BIOS_CONFIG (PMTA_MEE_BIOS_CONFIG)—Offset 4C18h

Shadow register for the Slicehash maintained in the MEE; Copy of the B_CR_SLICE_CHANNEL_HASH register (not used) with poison bit added at bit 5.

Access Method

Type: MEM Register
(Size: 64 bits)

Device:
Function:

Default: 3C00000000h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW/L	**LOCK (LOCK):** When set all writes to this register will be dropped
62:52	0h RO	**RESERVED_3 (RESERVED_3):** Reserved
51:38	0h RW/L	**CH_HASH_MASK (CH_HASH_MASK):** When both PMI channels in a slice are enabled this field specifies the Channel Hash Mask to be applied on Addr[19:6] postremap DRAM address of the request to compute which PMI channel a request must be routed to. Relevant only when HVM mode is disabled and only for requests that do not fall under the MOT region. Bunit will override the programmed value to include the Channel Selector bit. Additionally Bunit will remove the Slice Selector bit. See INTERLEAVE_MODE field. Note that HVM mode and MOT regions have special hash requirements and hence they do not use the CH_HASH_MASK.

Bit Range	Default & Access	Field Name (ID): Description
37:36	3h RW/L	**SYM_SLICE1_CHANNEL_ENABLED (SYM_SLICE1_CHANNEL_ENABLED):** Specifies which channels in Slice1 are enabled for hosting the symmetric region of the DRAM address space. If SLICE_1_DISABLED is set to disable mapping of DRAM address space to Slice 1, this field has no effect and symmetric region is considered to be not mapped to Slice1. If CH_1_DISABLED is set, Channel 1 in Slice 1 is not enabled for symmetric DRAM address space regardless of the setting of this register. DRAM address space is divided into a symmetric regions, asymmetric regions that are partially interleaved across a subset of channels (2-way) or asymmetric regions that are non-interleaved and mapped only to a single channel. This field specifies which channels in Slice 1 are enabled for interleaving the symmetric region. SLICE0_CHANNEL_ENABLED defines which channels in Slice 0 are enabled for interleaving the symmetric region. The mapping of partial (2-way) interleaved and non-interleaved regions are specified in other registers. For symmetric region, only selected configurations are supported. Across both slices, only a total of 1 channel, 2 channels or 4 channels must be enabled for interleaving the symmetric region. The total of 1, 2 or 4 channels can be made up of any combination of channel enables across both slices. B-unit does not support a 3-way interleaving of the symmetric region across 3 enabled channels in the two slices combined.
35:34	3h RW/L	**SYM_SLICE0_CHANNEL_ENABLED (SYM_SLICE0_CHANNEL_ENABLED):** Specifies which channels in Slice0 are enabled for hosting the symmetric region of the DRAM address space. If SLICE_0_MEM_DISABLED is set to disable mapping of DRAM address space to Slice 0, this field has no effect and symmetric region is considered to be not mapped to Slice0. If CH_1_DISABLED is set, Channel 1 in Slice 0 is not enabled for symmetric DRAM address space regardless of the setting of this register. DRAM address space is divided into a symmetric regions, asymmetric regions that are partially interleaved across a subset of channels (2-way) or asymmetric regions that are non-interleaved and mapped only to a single channel. This field specifies which channels in Slice 0 are enabled for interleaving the symmetric region. SLICE1_CHANNEL_ENABLED defines which channels in Slice 1 are enabled for interleaving the symmetric region. The mapping of partial (2-way) interleaved and non-interleaved regions are specified in other registers. For symmetric region, only selected configurations are supported. Across both slices, only a total of 1 channel, 2 channels or 4 channels must be enabled for interleaving the symmetric region. The total of 1, 2 or 4 channels can be made up of any combination of channel enables across both slices. B-unit does not support a 3-way interleaving of the symmetric region across 3 enabled channels in the two slices combined.
33	0h RO	**RESERVED_2 (RESERVED_2):** Resesved

Bit Range	Default & Access	Field Name (ID): Description
32	0h RW/L	**CH_1_DISABLED (CH_1_DISABLED):** Channel 1 in both slices are disabled no memory address mapped to ch 1. All requests sent to channel 0.
31	0h RW/L	**ENABLE_PMI_DUAL_DATA_MODE (ENABLE_PMI_DUAL_DATA_MODE):** When set to 1, Single Command Interface and Dual Data Interface for Reads and Writes
30:20	0h RO	**RESERVED_1 (RESERVED_1):** Reserved
19:6	0h RW/L	**SLICE_HASH_MASK (SLICE_HASH_MASK):** When both slices are enabled this field specifies the Slice Hash Mask to be applied on Addr[19:6] physical address of the request to compute which slice a request must be routed to. Relevant only when HVM mode is disabled and only for physical addresses that do not fall under the Asymmetric Memory Region and the MOT region. Bunit will override the programmed value to include the Slice Selector bit. Additionally Bunit will remove the Channel Selector bit. See INTERLEAVE_MODE field. Note that HVM mode nonaddress IDI requests asymmetric memory region and MOT regions have special hash requirements and hence they do not use the SLICE_HASH_MASK.
5	0h RW/L	**POISON (POISON):** Indication that Poison is enabled when set
4	0h RW/L	**SLICE_0_MEM_DISABLED (SLICE_0_MEM_DISABLED):** Slice 0 is disabled for memory accesses; no memory address mapped to Slice 0 and all memory requests sent to Slice 1.

Bit Range	Default & Access	Field Name (ID): Description
3:2	0h RW/L	**INTERLEAVE_MODE (INTERLEAVE_MODE):** Default interleave mode that specifies how the Slice Selector and Channel Selector bits are to be determined. Relevant only when HVM mode is disabled and only for system memory addresses that do not fall under the MOT region or the Asymmetric memory region in the System Address Map. Legal encodings are 0x0 0x1 and 0x2. An encoding of 0x3 is treated as if it was 0x2. When both slices and all four PMI channels are enabled 0x0: Default Slice Selector is Addr[10] and Default Channel Selector is Addr[11] 0x1: Default Slice Selector is Addr[11] and Default Channel Selector is Addr[12] 0x2: Default Slice Selector is Addr[12] and Default Channel Selector is Addr[13] When both slices are enabled but only one channel in each slice enabled: 0x0: Default Slice Selector is Addr[10] 0x1: Default Slice Selector is Addr[11] 0x2: Default Slice Selector is Addr[12] When only SLICE0 is enabled and both channels on that slice are enabled: 0x0: Default Channel Selector is Addr[10] 0x1: Default Channel Selector is Addr[11] 0x2: Default Channel Selector is Addr[12] When SLICE0 and only one channel in that slice is enabled this field is not relevant. Bunit overrides the setting of the SLICE_HASH_MASK to always include the Slice Selector bit. Similarly Bunit overrides the setting of the CH_HASH_MASK to always include the Channel Selector bit.
1	0h RW/L	**HVM_MODE (HVM_MODE):** 0: HVM mode is disabled. 1: HVM mode is enabled. When HVM mode is enabled Slice Hash and Channel Hash is done as follows: Both slices and all four PMI channels enabled: Slice Hash is Request Physical Addr[29] Channel Hash is PostRemap Addr[30] Both slices enabled but only one PMI channel in each slice enabled: Slice Hash is Request Physical Addr[29] Only one SLICE0 enabled but both PMI channels in SLICE0 enabled: Channel Hash is PostRemap Addr[29] When HVM_MODE is enabled TOLUD must be set at 2GB.
0	0h RW/L	**SLICE_1_DISABLED (SLICE_1_DISABLED):** Slice 1 is disabled no memory address mapped to Slice 1. All request sent to Slice 0

§ §

4 Processor Graphics Registers

This chapter documents the registers in Bus: 0, Device 2, Function 0.

NOTE: These registers apply to all processors.

4.1 B-Unit Copy of PCICMD for IGD (B_CR_PCICMD_0_2_0_PCI) — Offset 4h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:2, F:0] + 4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:11	0h RO	**Reserved (RESERVED_0):** Reserved
10	0h RW/V	**INTx Disable (INTDIS):** This bit disables the device from asserting INTx. • 0: Enable the assertion of this devices INTx signal. • 1: Disable the assertion of this devices INTx signal. DO_INTx messages will not be sent to DMI.
9	0h RO	**FB2B:** Not Implemented. Hardwired to 0.
8	0h RO	**SEN:** Not Implemented. Hardwired to 0.
7	0h RO	**WCC:** Not Implemented. Hardwired to 0.
6	0h RO	**PER:** Not Implemented. Hardwired to 0. Since the IGD belongs to the category of devices that does not corrupt programs or data in system memory or hard drives the IGD ignores any parity error that it detects and continues with normal operation.
5	0h RO	**VPS:** This bit is hardwired to 0 to disable snooping.
4	0h RO	**MWIE:** Hardwired to 0. The IGD does not support memory write and invalidate commands.
3	0h RO	**SCE:** This bit is hardwired to 0. The IGD ignores Special cycles.
2	0h RW/V	**BME:** • 0: Disable IGD bus mastering. • 1: Enable the IGD to function as a PCI compliant master. .
1	0h RW/V	**MAE:** This bit controls the IGD's response to memory space accesses. • 0: Disable. • 1: Enable. Device 2 Function Level Reset must reset this bit.
0	0h RW/V	**IOAE:** This bit controls the IGD's response to I/O space accesses. • 0: Disable. • 1: Enable. Device 2 Function Level Reset must reset this bit.

4.2 B-Unit Copy of GTTMMADR (B_CR_GTTMMADR_LO_0_2_0_PCI) — Offset 10h

This register contains the lower 32 bits of GTTMMADR.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:2, F:0] + 10h	4 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW/V	**Memory Base Address (MBA):** GTTMMADR[38:24] is {GTTMMADR_HI[6:0],GTTMMADR_LO[31:24]}. These bits, which are set by the OS, correspond to address signals [38:24]. 16MB combined for MMIO and Global GTT table aperture. 2MB are for MMIO, 6MB are reserved and 8 MB are for GTT.
23:4	0h RO	**Address Mask (ADM):** Hardwired to 0s to indicate at least 16MB address range.
3	0h RO	**Prefetch Memory Enable (PREFMEM):** Hardwired to 0 to prevent prefetching.
2:1	2h RO	**Memory Base Address Type (MEMTYP):** • 00: To indicate 32 bit base address • 01: Reserved • 10: To indicate 64 bit base address • 11: Reserved
0	0h RO	**Memory or IO Space (MIOS):** Hardwired to 0 to indicate memory space.

4.3 B-Unit Copy of GTTMMADR (B_CR_GTTMMADR_HI_0_2_0_PCI) — Offset 14h

This register contains the upper 32 bits of GTTMMADR.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:2, F:0] + 14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Reserved (RESERVED_0):** Reserved
6:0	0h RW/V	**Memory Base Address (MBA):** These bits are set by the OS, and correspond to address signals [38:24]. 16MB combined for MMIO and Global GTT table aperture. 2MB are for MMIO, 6MB are reserved and 8 MB are for GTT.

4.4 B-Unit Copy of GMADR (B_CR_GMADR_LO_0_2_0_PCI) — Offset 18h

This register contains the lower 32 bits of GMADR.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:2, F:0] + 18h	4 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/V	**Memory Base Address or Address Map 4096 (ADMSK4096):** This Bit is either part of the Memory Base Address R/W, or part of the Address Mask RO, depending on the value of MSAC[4:0].
30	0h RW/V	**Memory Base Address or Address Map 2048 (ADMSK2048):** This Bit is either part of the Memory Base Address R/W, or part of the Address Mask RO, depending on the value of MSAC[4:0].
29	0h RW/V	**Memory Base Address or Address Map 1024 (ADMSK1024):** This Bit is either part of the Memory Base Address R/W, or part of the Address Mask RO, depending on the value of MSAC[4:0].
28	0h RW/V	**Memory Base Address or Address Map 512 (ADMSK512):** This Bit is either part of the Memory Base Address R/W, or part of the Address Mask RO, depending on the value of MSAC[4:0].
27	0h RW/V	**Memory Base Address or Address Map 256 (ADMSK256):** This Bit is either part of the Memory Base Address R/W, or part of the Address Mask RO, depending on the value of MSAC[4:0].
26:4	0h RO	**Address Mask (ADM):** Hardwired to 0s to indicate at least 128MB address range.
3	0h RO	**Prefetch Memory Enable (PREFMEM):** Hardwired to 0 to prevent prefetching.
2:1	2h RO	**Memory Base Address Type (MEMTYP):** • 00: To indicate 32 bit base address • 01: Reserved • 10: To indicate 64 bit base address • 11: Reserved
0	0h RO	**Memory or IO Space (MIOS):** Hardwired to 0 to indicate memory space.

4.5 B-Unit Copy of GMADR (B_CR_GMADR_HI_0_2_0_PCI) — Offset 1Ch

This register contains the upper 32 bits of GMADR.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:2, F:0] + 1Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Reserved (RESERVED_0):** Reserved
6:0	0h RW/V	**Memory Base Address (MBA):** These bits are set by the OS, and correspond to address signals [38:32].

4.6 B-Unit Copy of the IOBAR (B_CR_IOBAR_0_2_0_PCI) — Offset 20h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:2, F:0] + 20h	1 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RESERVED_1):** Reserved
15:6	0h RW/V	**IO Base Address Register Select (IOBASE):** These bits correspond to address signals [15:6]. If there is a match then address hits the IOBAR. Device 2 FLR must reset this field.
5:3	0h RO	**Reserved (RESERVED_0):** Reserved
2:1	0h RO	**Memory Base Address Type (MEMTYPE):** Hardwired to 0s to indicate 32bit address.
0	1h RO	**Memory or IO Space (MIOS):** Hardwired to 1 to indicate IO space.

4.7 B-Unit Copy of Device 2 Control Register (B_CR_DEV2CTL_0_2_0_PCI) — Offset 58h

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:2, F:0] + 58h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:1	0h RO	**Reserved (RESERVED_0):** Reserved
0	0h RW/V	**IOBAR Disable (IOBARDIS):** System BIOS can choose to disable and hide the IOBAR for systems that do not require legacy IOBAR access to Gfx MMIO registers. • 0: IOBAR is enabled and exposed at offset 0x20 in Device 2 Configuration space. Default • 1: IOBAR is disabled and not visible in PCI Configuration Space. Behaves as if hardwired to zeros.

4.8 B-Unit Copy of MSAC (B_CR_MSAC_0_2_0_PCI) — Offset 62h

This data is used to determine the size of Aperture GMADR, and affects certain bits of the GMADR register.

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:2, F:0] + 62h	1 h

Bit Range	Default & Access	Field Name (ID): Description
7:5	0h RO	**Reserved (RESERVED_0):** Reserved
4	0h RW/V	**Aperture Size 4 (APSZ4):** Refer to description for APSZ0 [0:0].
3	0h RW/V	**Aperture Size 3 (APSZ3):** Refer to description for APSZ0 [0:0].
2	0h RW/V	**Aperture Size 2 (APSZ2):** Refer to description for APSZ0 [0:0].
1	0h RW/V	**Aperture Size 1 (APSZ1):** Refer to description for APSZ0 [0:0].
0	1h RW/V	**Aperture Size 0 (APSZ0):** This field is used in conjunction with other APSZ fields to determine the size of Aperture GMADR. It affects certain bits of the GMADR register. The description below is for all APSZ fields [4:0]: • 00000b: 128MB GMADR.B[26:4] is hardwired to 0. • 00001b: 256MB GMADR.B[27]=0, RO. • 00010b: illegal hardware will treat this as 00011. • 00011b: 512MB GMADR.B[28:27]=0, RO. • 00100b-00110b: illegal hardware will treat this as 00111. • 00111b: 1024MB GMADR.B[29:27]=0,RO. • 01000b-01110b: illegal hardware will treat this as 01111. • 01111b: 2048MB GMADR.B[30:27]=0,RO. • 10000b-11110b: illegal hardware will treat this as 11111. • 11111b: 4096MB GMADR.B[31:27]=0,RO

4.9 B-Unit Copy of Device 2 Control Register (B_CR_DEVICECTL_0_2_0_PCI) — Offset 78h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:2, F:0] + 78h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/V	**Initiate Function Level Reset (INITIATE_FLR):** A write of 1b initiates Function Level Reset to the Function. The value read by software from this bit is always 0b. B-Unit uses only this bit for Dev2 FLR. The rest of the bits in this register are ignored by the B-Unit.
14:12	0h RO	**Max Read Request Size (MAX_READ_REQUEST_SIZE):** Functions that do not generate Read Requests larger than 128 bytes and Functions that do not generate Read Requests on their own behalf are permitted to implement this field as Read Only RO with a value of 000b.
11	0h RO	**Enable No Snoop (ENABLE_NO_SNOOP):** This bit is permitted to be hardwired to 0b if a function would never set the No Snoop attribute in transactions it initiates. The graphics device never generates a PCI Express TLP.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RO	**Aux Power PM Enable (AUX_PM_ENABLE):** Functions that do not implement this capability hardwire this bit to 0b.
9	0h RO	**Phantom Functions Enable (PHANTOM_FUNCTIONS_ENABLE):** Functions that do not implement this capability hardwire this bit to 0b.
8	0h RO	**Extended Tag Field Enable (EXTENDED_TAG_ENABLE):** Functions that do not implement this capability hardwire this bit to 0b.
7:5	0h RO	**Max Payload Size (MAX_PAYLOAD_SIZE):** Functions that support only the 128byte max payload size are permitted to hardwire this field to 000b.
4	0h RO	**Enable Relaxed Ordering (RO_ENABLE):** A Function is permitted to hardwire this bit to 0b if it never sets the Relaxed Ordering attribute in transactions it initiates as a Requester. The graphics device never generates a PCI Express TLP.
3	0h RO	**Unsupported Request Response Enable (UR_ENABLE):** A Root Complex Integrated Endpoint that is not associated with a Root Complex Event Collector is permitted to hardwire this bit to 0b.
2	0h RO	**Fatal Error Enable (FATAL_ERR_ENABLE):** A Root Complex Integrated Endpoint that is not associated with a Root Complex Event Collector is permitted to hardwire this bit to 0b.
1	0h RO	**NonFatal Error Enable (NONFATAL_ERR_ENABLE):** A Root Complex Integrated Endpoint that is not associated with a Root Complex Event Collector is permitted to hardwire this bit to 0b.
0	0h RO	**Correctable Error Enable (CORRECTABLE_ERR_ENABLE):** A Root Complex Integrated Endpoint that is not associated with a Root Complex Event Collector is permitted to hardwire this bit to 0b.

4.10 B-Unit Copy of PMCS for IGD (B_CR_PMCS_0_2_0_PCI) — Offset D4h

This is the B-Unit Copy of the Power Management Control/Status Register for the Internal Graphics Device.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:2, F:0] + D4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**Power Management Event Status (PMESTS):** This bit is 0 to indicate that the Internal Graphics Device does not support Power Management Event generation from D3 (cold).
14:13	0h RO	**Data Scale (DSCALE):** The Internal Graphics Device does not support data register. This bit always returns 00 when read write operations have no effect.
12:9	0h RO	**Data Select (DSEL):** The IGD does not support data register. This bit always returns 0h when read write operations have no effect.
8	0h RO	**Power Management Event Enable (PMEEN):** This bit is 0 to indicate that PME assertion from D3 (cold) is disabled.
7:2	0h RO	**Reserved (RESERVED_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW/V	**Power State (PWRSTAT):** This field indicates the current power state of the IGD, and can be used to set the IGD into a new power state. If software attempts to write an unsupported state to this field, the write operation must complete normally on the bus, but the data is discarded and no state change occurs. On a transition from D3 to D0, the graphics controller is optionally reset to initial values. • 00: D0 Default • 01: D1 Not Supported • 10: D2 Not Supported • 11: D3

§ §

5 System Agent Address Map Registers

This chapter documents the registers in Bus: 0, Device 0, Function 1.

5.1 Device ID and Vendor ID Register (P_CR_DEVICE_ID_VENDOR_ID_0_0_1_PCI) — Offset 0h

This register uniquely identifies any PCI device.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:1] + 0h	A8C8086 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	A8Ch RO/V	**Device ID (DEVICE_ID):** Identifier assigned to the Broxton Thermal Management Controller.
15:0	8086h RO	**Vendor ID (VENDOR_ID):** Hardwired to Intel's Vendor ID value.

5.2 PCI_STATUS_COMMAND_0_0_1_PCI (B_CR_PCI_STATUS_COMMAND_0_0_1_PCI) — Offset 4h

PCI Status is used to record status information for PCI bus related events, including the occurrence of a PCI compliant Master Abort (MA) and PCI compliant Target Abort (TA). PCISTS also indicates the DEVSEL# timing that has been set by the thermal device.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:1] + 4h	900000 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Detected Parity Error (DPE):** The Thermal device does not implement this bit and it is hardwired to a 0.
30	0h RO	**Signaled System Error (SSE):** This bit is hardwired to zero. The Thermal device never asserts SERR#, and therefore it has no need to implement this bit.
29	0h RO	**Received Master Abort (RURS):** The Thermal device does not implement this bit and it is hardwired to a 0.
28	0h RO	**Received Target Abort (RCAS):** The Thermal device does not implement this bit and it is hardwired to a 0.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**Signaled Target Abort (STAS):** his bit is hardwired to 0. The Thermal device will not generate a Target Abort DMI completion packet or Special Cycle, and therefore it has no need to implement this bit.
26:25	0h RO	**DEVSEL Timing (DEVT):** These bits are hardwired to 0. Device 4 does not physically connect to PCI_A.
24	0h RO	**Master Data Parity Error (DPD):** This bit is hardwired to 0. PERR signaling and messaging are not implemented by the Thermal Device, and therefore it has no need to implement this bit.
23	1h RO	**Fast Back to Back Capable (FB2B_CAPABLE):** This bit is hardwired to 1. Device 4 does not physically connect to PCI_A, so this bit is set to 1 (indicating fast back-to-back capability) so that the optimum setting for PCI_A is not limited by the Thermal device.
22	0h RO	**Reserved (RSVD):** Reserved
21	0h RO	**66MHz Capable (PCI66M):** The Thermal device does not implement this bit and it is hardwired to a 0.
20	1h RO	**Capabilities List (CLIST):** This bit is set to 1 to indicate that the register at 34h provides an offset into the function. PCI Configuration Space containing a pointer to the location of the first item in the list.
19	0h RO/V	**Interrupt Status (IS):** Reflects the state of the INTA# signal at the input of the enable/disable circuit. This bit is set by HW to 1 when the INTA# is asserted and reset by HW to 0 after the interrupt is cleared (independent of the state of the Interrupt Disable bit in the 0.0.1.PCICMD register).
18:11	0h RO	**Reserved (RSVD):** Reserved
10	0h RW	**INTx# Disable (INTDIS):** This bit, when set, disables the device from asserting INTx#, where 'x' is configured in the INTPIN register.
9	0h RO	**Fast Back to Back Enable (FB2B_ENABLE):** The Thermal device does not implement this bit and it is hardwired to a 0.
8	0h RO	**SERR# Enable (SERRE):** The Thermal device does not implement this bit and it is hardwired to a 0.
7	0h RO	**ADSTEP:** The Thermal device does not implement this bit and it is hardwired to a 0.
6	0h RO	**Parity Error Enable (PERRE):** This bit is hardwired to 0. The Thermal device belongs to the category of devices that does not corrupt programs or data in system memory or hard drives. It therefore ignores any parity error that it detects and continues with normal operation.
5	0h RO	**VGA Snoop (VGASNOOP):** The Thermal device does not implement this bit and it is hardwired to a 0.
4	0h RO	**Memory Write and Invalidate Enable (MWIE):** This bit is hardwired to 0. The Thermal device will never issue memory write and invalidate commands, and therefore has no need to implement this bit.
3	0h RO	**Special Cycle Enable (SCE):** The Thermal device does not implement this bit and it is hardwired to a 0.
2	0h RW	**Bus Master Enable (BME):** The Thermal device is enabled to function as a PCI-compliant bus master when this bit is set. If it is not set, bus mastering is disabled.
1	0h RW	**Memory Access Enable (MAE):** The Thermal device will allow access to thermal registers when this bit is set. If it is not set, access to memory mapped thermal registers is disabled.
0	0h RO	**I/O Access Enable (IOAE):** The Thermal device does not implement this bit and it is hardwired to a 0.

5.3 PCI Revision ID and PCI Class Code Register (P_CR_REVISION_ID_CLASS_CODE_0_0_1_PCI) — Offset 8h

Revision ID contains the revision number of the device. Class Code identifies the basic function of the device.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:1] + 8h	11800000 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	11h RO	**Base Class Code (BASE_CLASS_CODE):** This is an 8-bit value that indicates the base class code for the Thermal Controller. This code has the value 11h, indicating a device that is used for data acquisition and signal processing.
23:16	80h RO	**Sub-class Code (SUB_CLASS_CODE):** The code is 80h which indicates 'Other Data Acquisition and Signal Processing Controllers.'
15:8	0h RO	**Programming Interface (PI):** This is an 8-bit value that indicates the programming interface of this device. This value does not specify a particular register set layout and provides no practical use for this device.
7:0	0h RO/V	**Revision ID (REVISION_ID):** Revision ID.

5.4 Master Latency Timer and Header Type Register (P_CR_MASTER_LATENCY_TIME_0_0_1_PCI) — Offset Ch

This register defines Latency Timer and layout of the device Configuration Space header.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:1] + Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**BIST Support (BS):** This bit is hardwired to zero. The Thermal device does not support BIST.
30:24	0h RO	**Reserved (RSVD):** Reserved
23:16	0h RO	**Header Type (HDR):** This field always returns 0 to indicate that the Thermal device is a single function device with standard header layout.
15:8	0h RO	**Master Latency Timer (MLT):** This field is hardwired to 0. The Thermal device does not support perform bursts.
7:0	0h RO	**Cache Line Size (CLS):** This field is hardwired to 0. The Thermal Device as a PCI compliant master does not use the Memory Write and Invalidate command and, in general, does not perform operations based on cache line size.

5.5 Thermal Management Base Address Register (B_CR_TMBAR_LO_0_0_1_PCI) — Offset 10h

This is the base address for the Thermal Controller Memory Mapped space. There is no physical memory within this 32KB window that can be addressed. The 32KB reserved by this register does not alias to any PCI 2.2 compliant memory mapped space. All TMBAR space maps the access to this memory space towards MCHBAR space. For details of this BAR, refer to the MCHBAR specifications.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:1] + 10h	4 h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RW	**TMMBA:** This field corresponds to bits 31 to 15 of the base address TMBAR address space. BIOS will program this register resulting in a base address for a 32KB block of contiguous memory address space. This register ensures that a naturally aligned 32KB space is allocated within total addressable memory space. The DPTF driver uses this base address to program all Thermal and Throttling control register set.'
14:4	0h RO	**Address Map (ADM):** Hardwired to 11'h000 to indicate at least 32KB address range.
3	0h RO	**Prefetch Memory (PM):** Hardwired to 1'b0 to prevent prefetching.
2:1	2h RO	**Memory Type (MT):** Hardwired to 2'b10 to indicate 64-bit address.
0	0h RO	**Memory or IO Space (MIOS):** Hardwired to 0 to indicate memory space.

5.6 Thermal Management Base Address Register (B_CR_TMBAR_HI_0_0_1_PCI) — Offset 14h

This is the base address for the Thermal Controller Memory Mapped space. There is no physical memory within this 32KB window that can be addressed. The 32KB reserved by this register does not alias to any PCI 2.2 compliant memory mapped space. All TMBAR space maps the access to this memory space towards MCHBAR space. For details of this BAR, refer to the MCHBAR specifications.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:1] + 14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RW	**Reserved RW (RSVDRW):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
6:0	0h RW	**TMMBA:** This field corresponds to bits 38 to 32 of the base address TMBAR address space. BIOS will program this register resulting in a base address for a 32KB block of contiguous memory address space. This register ensures that a naturally aligned 32KB space is allocated within total addressable memory space. The DPTF driver uses this base address to program all Thermal and Throttling control register set.

5.7 PCI Subsystem Vendor ID and PCI Subsystem ID (P_CR_SVID_SID_0_0_1_PCI) — Offset 2Ch

This register is used to uniquely identify the subsystem where the PCI device resides.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:1] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Subsystem ID (SUBSYSTEM_ID):** PCI Subsystem ID: This field should be programmed during BIOS initialization.
15:0	0h RW	**Subsystem Vendor ID (SUBSYSTEM_VENDOR_ID):** PCI Subsystem Vendor ID: This field should be programmed by BIOS during bootup to indicate the vendor of the system board.

5.8 CAPPTR_0_0_1_PCI (P_CR_CAPPTR_0_0_1_PCI) — Offset 34h

CAPPOINT provides the offset that is the pointer to the location of the first device capability in the capability list.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:1] + 34h	D0 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD):** Reserved
7:0	D0h RO	**Base Address of First Capability Register (BASE_ADDR):** This pointer is an 8-bit address to an offset within this device's Configuration Space that holds the first Capability Register (CAPID0_CAPCTRL).

5.9 Interrupt and Latency Configuration (P_CR_INTR_LAT_0_0_1_PCI) — Offset 3Ch

This register is used for specifying how often the device needs to gain access to the PCI bus.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:1] + 3Ch	100 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Maximum Latency Value (MLV):** These bits are hardwired to zero. The Thermal device has no specific requirements for how often it needs to access the PCI bus.
23:16	0h RO	**Minimum Grant Value (MGV):** These bits are hardwired to zero. The Thermal device does not burst as a PCI compliant master.
15:8	1h RW/L	**Interrupt Pin (INTPIN):** As a single function device, the Thermal device specifies INTA as its interrupt pin. 01h = INTA by default. BIOS may need to reconfigure this in order allow software to disambiguate multiple functions within MCHBAR. Recommended setting is 02h = INTB. This field is locked from future writes when the THERMAL_DEVICE_IRQ.LOCK
7:0	0h RW	**Interrupt Line (INTCON):** Used to communicate interrupt line routing information. BIOS Requirement: POST software writes the routing information into this register as it initializes and configures the system. The value indicates to which input of the system interrupt controller this device's interrupt pin is connected.

5.10 Device Enable Register (P_CR_DEVEN_0_0_1_PCI) — Offset 54h

The Device Enable register allows for enabling/disabling of PCI devices and functions that are within the CPU package.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:1] + 54h	33 h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RO	**RESERVED_1:** Reserved
5	1h RW/V	**Device 3 Function 0 Enable (Imaging) (D3F0EN):** • 0 = Imaging Device 0/3/0 is disabled and hidden • 1 = Imaging Device 0/3/0 is enabled and visible Software may write a 0b to this register to disable the imaging device. Writes to 1b are not allowed (i.e., the device may not be software enabled if hardware natively disables it)
4	1h RW/V	**Device 2 Function 0 Enable (Graphics/Display) (D2F0EN):** • 0 = Graphics/Display Device 0/2/0 is disabled and hidden • 1 = Graphics/Display Device 0/2/0 is enabled and visible Software may write a 0b to this register to disable the graphics / display device. Writes to 1b are not allowed (i.e., the device may not be software enabled if hardware natively disables it)
3	0h RO	**Reserved (RSVD):** Reserved
2	0h RO	**RESERVED_0:** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1	1h RW/V	**Device 0 Function 1 Enable (Thermal) (D0F1EN):** • 0 = Thermal Device 0/0/1 is disabled and hidden • 1 = Thermal Device 0/0/1 is enabled and visible Software may write a '0 to this register to disable the thermal device. Writes to '1 are not allowed (i.e., the device may not be software enabled if hardware natively disables it)
0	1h RO	**Device 0 Function 0 Enable (Memory / System Agent) (D0F0EN):** Device enable for MCHBAR device. This device not be disabled and is therefore hardwired to 1.

5.11 SCISTS_0_0_1_PCI (P_CR_SCISTS_0_0_1_PCI) — Offset 88h

This register is used to report various error conditions via the SCI messaging mechanism. An SCI message is generated on a zero to one transition of any of these flags (if enabled by the SCICMD and PCICMD registers). This bit is set regardless of whether or not the SCI event is enabled and generated. After the event processing is complete, the error logging mechanism can be unlocked by clearing the appropriate status bit by software writing a 1 to it.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:1] + 88h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	**Reserved (RSVD):** Reserved
0	0h RW/1C/V	**CTED:** If set, indicates a thermal event was detected and SCI has optionally been generated. If this bit is already set, then an interrupt message will not be sent on a new event. ACPI software will write this bit to 1b to clear the event.

5.12 SCI Command (P_CR_SCICMD_0_0_1_PCI) — Offset CCh

This register enables various event conditions to generate an SCI event message. When an event log is observed, it will generate an SCI message when enabled by the SCICMD register.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:1] + CCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW	**SCI Event Enable (SCICE):** • 0 = SCI messaging in response to thermal events is disabled. • 1 = Generate an SCI event message on detection of a hardware thermal event, as configured by software.
15:0	0h RO	**Reserved (RSVD):** Reserved

5.13 Power Management Capabilities (P_CR_PMCAPID_0_0_1_PCI) — Offset D0h

The Power Management Capabilities register is a 16-bit read-only register which provides information on the capabilities of the function related to power management. The information in this register is generally static and known at design time.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:1] + D0h	3E001 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	0h RO	**PME Support (PMES):** This field indicates the power states in which the thermal device may assert PME#. It is hardwired to 0 to indicate that the Thermal device does not support nor assert the PME# signal.
26	0h RO	**D2:** Hardwired to 0 to indicate that the D2 power management state is not supported.
25	0h RO	**D1:** Hardwired to 0 to indicate that the D1 power management state is not supported.
24:22	0h RO	**Reserved (RSVD):** Reserved
21	0h RO	**Device Specific Initialization (DSI):** Indicates whether special initialization of this function is required (beyond the standard PCI configuration header) before the generic class device driver is able to use it. This bit is not used by some operating systems. Microsoft Windows* and Windows NT, for instance, do not use this bit to determine whether to use D3. Instead, they use the driver's capabilities to determine this. 1b indicates that the function requires a device specific initialization sequence following transition to the D0 uninitialized state.
20	0h RO	**APS:** The Thermal device does not implement this bit and it is hardwired to a 0.
19	0h RO	**PME# Capbility (PMEC):** Hardwired to 0 to indicate the thermal device does not support PME# generation.
18:16	3h RO	**VER:** This device complies with revision 1.2 of the PCI Power Management Interface Specification.
15:8	E0h RO	**Next Capability Pointer (NCP):** This contains a pointer to next item in capabilities list. The next capability is E0h which is device 0 / func 0 capability mirror.
7:0	1h RO	**Capability ID (CID):** 01h indicates that this is a power management capability

5.14 Power Management Control and Status (P_CR_PMCS_0_0_1_PCI) — Offset D4h

The Data register is an optional, 8-bit read-only register that provides a mechanism for the function to report state dependent operating data such as power consumed or heat dissipation. Typically the data returned through the Data register is a static copy (look up table, for example) of the function's worst case 'DC characteristics' data sheet. This data, when made available to system software, could then be used to intelligently make decisions about power budgeting, cooling requirements, etc.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:1] + D4h	8 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**DATA:** The data register, data scale and data select registers are not supported. Hardwired to zero.
23:16	0h RO	**Reserved (RSVD):** Reserved
15	0h RO	**PME# Status (PMES):** This bit is set when the function would normally assert the PME# signal independent of the state of the PME_En bit. This bit is hardwired to 0b to indicate that PME# assertion from D3 (cold) is not supported.
14:13	0h RO	**Data Scale (DS):** The data register, data scale and data select registers are not supported
12:9	0h RO	**Data Select (DSEL):** The data register, data scale and data select registers are not supported
8	0h RO	**PME# Enable (PMEEN):** This bit is hardwired to 0b to indicate that PME# assertion from D3 (cold) is disabled.
7:4	0h RO	**Reserved (RSVD):** Reserved
3	1h RO	**No Soft Reset (NSR):** • 1 = Devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset. Configuration context is preserved. Upon transition from the D3hot to the D0 initialized state, no additional operating system intervention is required to preserve configuration Context beyond writing the PowerState bits. • 0 = Devices do perform an internal reset upon transitioning from D3hot to D0 via software control of the PowerState bits. Configuration context is lost when performing the soft reset. Upon transition from the D3hot to the D0 state, full initialization sequence is needed to return the device to D0 initialized. Regardless of this bit, devices that transition from D3hot to D0 by a system or bus segment reset will return to the device state D0 uninitialized with only PME context preserved if PME is supported and enabled.
2	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW/V	**Power State (PS):** This field indicates the current power state of the thermal device and can be used to set the thermal device into a new power state. If software attempts to write an unsupported state to this field, the write operation must complete normally on the bus, but the data is discarded and no state change occurs. • 00b = D0 Default • 01b = D1 Not Supported • 10b = D2 Not Supported • 11b = D3

5.15 Interrupt Status (P_CR_INTSTS_0_0_1_PCI) — Offset DCh

This register is used to report interrupt pending event status. Upon event detection, if enabled, an interrupt message will be delivered to the IOAPIC. Upon processing that event, software may clear the event to enable future event signaling by writing a 1 to the INTSTAT bit in this register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:1] + DCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	**Reserved (RSVD):** Reserved
0	0h RW/1C/V	**Interrupt Status (INTSTAT):** If set, indicates a thermal event was detected. If this bit is already set, then an interrupt message will not be sent upon detection of a new event.

5.16 Capability ID0 Capability Control (P_CR_CAPID0_CAPCTRL0_0_0_1_PCI) — Offset E0h

Control bits in this register describe the attributes of CAPID0_A and CAPID0_B capability registers.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:1] + E0h	10C0009 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD):** Reserved
27:24	1h RO	**Capability ID Version (CAPID_VER):** This field has the value 0001b to identify the first revision of the CAPID register definition.
23:16	Ch RO	**Capability ID0 Structure Length (CAPIDLEN):** This field has the value 0Ch to indicate the structure length 12 bytes. This is the total size of this CAPCTRL and the CAPID0_A and CAPID0_B registers in the following bytes.

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	**Next Capability Register Pointer (NCP):** This field is hardwired to 00h, indicating the end of the capabilities linked list.
7:0	9h RO	**Capability ID (CAP_ID):** This field has the value 1001b to identify the CAP_ID assigned by the PCI SIG for vendor dependent capability pointers.

5.17 Capability ID0 A (P_CR_CAPID0_A_0_0_1_PCI) — Offset E4h

Control of bits in this register is only required for SKU differentiation.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:1] + E4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW/V	**SPARE31_24:** Reserved for future capabilities.
23	0h RW/V	**VTd Disable (VTDD):** • 0 = Enable VTd • 1 = Disable VTd
22	0h RW/V	**FUSE_SPARE22:** Fuse backed spare.
21	0h RW/V	**FUSE_SPARE21:** Fuse backed spare.
20	0h RW/V	**FUSE_SPARE20:** Fuse backed spare.
19	0h RW/V	**FUSE_SPARE19:** Fuse backed spare.
18	0h RW/V	**FUSE_SPARE18:** Fuse backed spare.
17	0h RW/V	**FUSE_SPARE17:** Fuse backed spare.
16	0h RW/V	**FUSE_SPARE16:** Fuse backed spare.
15	0h RW/V	**Thermal Device Disable (CDD):** • 0 = Thermal associated memory spaces are accessible. • 1 = Thermal associated memory and IO spaces are disabled. DEVEN_0_0_0_PCI field for DPTF cannot be set.
14	0h RW/V	**FUSE_SPARE14:** Fuse backed spare.
13	0h RW/V	**FUSE_SPARE13:** Fuse backed spare.
12	0h RW/V	**FUSE_SPARE12:** Fuse backed spare.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW/V	**Internal Graphics and Display Disable (IGD):** • 0 = There is a graphics engine within this CPU. Internal Graphics Device 2 is enabled, and all of its memory and I/O spaces are accessible. Configuration cycles to Device 2 will be completed within the CPU. All nonSMM memory and IO accesses to VGA will be handled based on Memory and IO enables of Device 2, IO registers within Device 2, and VGA Enable of the PCI to PCI bridge control register in Devices 1 and 6, if PCI Express GFX attach is supported. A selected amount of Graphics Memory space is preallocated from the main memory, based on Graphics Mode Select GMS in the GGC Register. Graphics Memory is preallocated above TSEG Memory. • 1 = There is no graphics engine within this CPU. Internal Graphics Device 2 and all of its memory and I/O functions are disabled. Configuration cycles targeted to Device 2 will be passed on to DMI. In addition, all clocks to internal graphics logic are turned off. All nonSMM memory and IO accesses to VGA will be handled based on VGA Enable of the PCI to PCI bridge control register in Devices 1 and 6. DEVEN [4:3] Device 0 offset 54h have no meaning. Device 2 Functions 0 and 1 are disabled and hidden.
10	0h RO	**DIDOE:** Controls if there is an override of Dev2 (GFX) device ID. Hardwired to 1'b0. • 0 = Disable ability to override DID - For production • 1 = Enable ability to override DID - For debug and samples only
9:8	0h RO	**CDID:** Controls the value of Dev2 (GFX) device ID. Hardwired to 2'b00. Identifier assigned to the core/primary PCI device.The corresponding two bit capability ID programming is: • 00b = Desktop • 01b = Mobile • 10b = Server • 11b = Marketing Spare
7:0	0h RW	**SPARE7_0:** Reserved for future capabilities

5.18 Capability ID0 B (P_CR_CAPID0_B_0_0_1_PCI) — Offset E8h

Control of bits in this register is only required for SKU differentiation.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:1] + E8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/V	**IMGU_DIS:** 0: Imaging Unit associated memory spaces are accessible. 1: Imaging Unit associated memory and IO spaces are disabled. DEVEN_0_0_0_PCI field for I-unit can not be set.
30	0h RW/V	**FUSE_SPARE30:** Fuse backed spare.
29	0h RW/V	**FUSE_SPARE29:** Fuse backed spares, potentially to be used for PKGTYP encoding.
28	0h RW/V	**FUSE_SPARE28:** Fuse backed spares potentially to be used for PKGTYP encoding.
27	0h RW/V	**FUSE_SPARE27:** Fuse backed spares potentially to be used for PKGTYP encoding.

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW/V	**FUSE_SPARE26:** Fuse backed spares potentially to be used for PKGTYP encoding.
25	0h RW/V	**FUSE_SPARE25:** Fuse backed spares potentially to be used for PKGTYP encoding.
24	0h RW/V	**Shared Virtual Memory Disable (SVMDIS):** • 0 = Enable Shared Virtual Memory mode • 1 = Disable Shared Virtual Memory mode
23:0	0h RW	**Spare (SPARE23_0):** Reserved for future capabilities.

5.19 B-Unit Copy of Default VTd BAR PMEN (B_CR_PMEN_REG_0_0_0_DEFVTDBAR)—Offset 64h

B-Unit copy of the Default VTd BAR PMEN register. Bit 31 is the only bit used for B-Unit.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/V	**EPM (EPM):** This field controls DMA accesses to the protected lowmemory and protected highmemory regions. • 0: Protected memory regions are disabled. • 1: Protected memory regions are enabled. Access control is implemented. IA accesses are always allowed access. NonIA accesses are allowed if request SAI matches the SAI of the Default VTd Engine. Requests blocked due to protected memory region violation are not recorded or reported as remapping faults. Hardware reports the status of the protected memory enable/disable operation through the PRS field in this register. Hardware implementations supporting DMA draining must drain any inflight translated DMA requests queued within the RootComplex before indicating the protected memory region as enabled through the PRS field.
30:1	0h RO	**Reserved (RESERVED_0):** Reserved.
0	0h RO	**PRS Unused (PRS):** Unused by the B-Unit. This field indicates the status of protected memory regions: • 0: Protected memory regions are disabled. • 1: Protected memory regions are enabled.

5.20 B-Unit Copy of Default VTd BAR PLM Base Register (B_CR_PLMBASE_REG_0_0_0_DEFVTDBAR)—Offset 68h

Register to set up the base address of DMA-protected low-memory region below 4GB. This register must be set up before enabling protected memory through PMEN_REG, and must not be updated when protected memory regions are enabled. This register is always treated as RO for implementations not supporting protected low memory region (PLMR field reported as Clear in the Capability register). The alignment of the protected low memory region base depends on the number of reserved bits (N:0) of this register. Software may determine N by writing all 1s to this register, and finding the most

significant bit position with 0 in the value read back from the register. Bits N:0 of this register are decoded by hardware as all 0s. Software must setup the protected low memory region below 4GB. Section 10.4.18 of the Intel Virtualization Technology for Directed I/O: Spec describes the Protected Low-Memory Limit register and hardware decoding of these registers. Software must not modify this register when protected memory regions are enabled (PRS field Set in PMEN_REG).

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW/V	**Protected Low Memory Base (PLMB):** This register specifies the base of the protected low-memory region in system memory.
19:0	0h RO	**Reserved (RESERVED_0):** Reserved.

5.21 B-Unit Copy of Default VTd BAR PLM Limit Register (B_CR_PLMLIMIT_REG_0_0_0_DEFVTDBAR)—Offset 6Ch

This register sets up the limit address of DMA-protected low-memory region below 4GB. This register must be set up before enabling protected memory through PMEN_REG, and it must not be updated when protected memory regions are enabled. This register is always treated as RO for implementations not supporting protected low memory region (PLMR field reported as Clear in the Capability register). The alignment of the protected low memory region limit depends on the number of reserved bits (N:0) of this register. Software may determine N by writing all 1's to this register, and then finding most significant zero bit position with 0 in the value read back from the register. Bits N:0 of the limit register are decoded by hardware as all 1s. The Protected low-memory base and limit registers function as follows: Programming the protected low-memory base and limit registers with the same value in bits 31:(N+1) specifies a protected low-memory region of size 2^(N+1) bytes. Programming the protected low-memory limit register with a value less than the protected low-memory base register disables the protected low-memory region. Software must not modify this register when protected memory regions are enabled (PRS field Set in PMEN_REG).

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW/V	**Protected Low Memory Limit (PLML):** This register specifies the last host physical address of the DMAprotected lowmemory region in system memory.

Bit Range	Default & Access	Field Name (ID): Description
19:0	0h RO	**Reserved (RESERVED_0):** Reserved.

5.22 B-Unit Copy of Default VTd BAR PHM Base Register (B_CR_PHMBASE_REG_0_0_0_DEFVTDBAR)—Offset 70h

This is the register to set up the base address of the DMA-protected high-memory region. This register must be set up before enabling protected memory through PMEN_REG, and must not be updated when protected memory regions are enabled. This register is always treated as RO for implementations not supporting protected high memory region (PHMR field reported as Clear in the Capability register). The alignment of the protected high memory region base depends on the number of reserved bits (N:0) of this register. Software may determine N by writing all 1's to this register, and finding most significant zero bit position below host address width (HAW) in the value read back from the register. Bits N:0 of this register are decoded by hardware as all 0s. Software may setup the protected high memory region either above or below 4GB. Software must not modify this register when protected memory regions are enabled (PRS field Set in PMEN_REG).

Access Method

Type: MEM Register (Size: 64 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_1):** Reserved.
38:20	0h RW/V	**Protected High Memory Base (PHMB):** This register specifies the base of protected high memory region in system memory. Hardware ignores and does not implement bits 63:HAW, where HAW is the host address width.
19:0	0h RO	**Reserved (RESERVED_0):** Reserved.

5.23 B-Unit Copy of Default VTd BAR PHM Limit Register (B_CR_PHMLIMIT_REG_0_0_0_DEFVTDBAR)—Offset 78h

Register to set up the limit address of DMA-protected high-memory region. This register must be set up before enabling protected memory through PMEN_REG, and must not be updated when protected memory regions are enabled. This register is always treated as RO for implementations not supporting protected high memory region (PHMR field reported as Clear in the Capability register). The alignment of the protected high memory region limit depends on the number of reserved bits (N:0) of this register. Software may determine the value of N by writing all ones to this register, and finding most significant zero bit position below host address width (HAW) in the value read back from the register. Bits N:0 of the limit register are decoded by hardware as all ones. The protected high-memory base and limit registers function as

follows: Programming the protected low-memory base and limit registers with the same value in bits HAW:(N+1) specifies a protected low-memory region of size 2^(N+1) bytes. Programming the protected high-memory limit register with a value less than the protected high-memory base register disables the protected high-memory region. Software must not modify this register when protected memory regions are enabled (PRS field Set in PMEN_REG).

Access Method

Type: MEM Register (Size: 64 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:39	0h RO	**Reserved (RESERVED_1):** Reserved.
38:20	0h RW/V	**Protected High Memory Limit (PHML):** This register specifies the last host physical address of the DMA protected high memory region in system memory. Hardware ignores and does not implement bits 63:HAW where HAW is the host address width.
19:0	0h RO	**Reserved (RESERVED_0):** Reserved.

§ §

6 Gaussian Mixture Model (GMM) Registers

This chapter documents the registers in Bus: 0, Device 0, Function 3.

6.1 ID — Offset 0h

Device ID assigned to GNA and Vendor ID

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:3] + 0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO/V	**Device Identification Number (DID):** Indicates the device ID assigned to the GNA [6:0] set by straps [15:7] This field is set to the module via IOSF SB message received during device reset.
15:0	0h RO	**Vendor Identification Number (VID):** Intels identification

6.2 Device Control (DCTRL) — Offset 4h

The Command register provides coarse control over GMM's abilities like Unsupported Request Error Reporting Enable, Poisoned TLP Error Reporting Enable, Interrupt Disable, Max Aligned Payload Size, Max Aligned Read Request Size, Special Cycle Enable, Bus Master Enable, Memory Space Enable

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:0, F:3] + 4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**Reserved (RSVD):** Reserved
14	0h RO	**Unsupported Request Error Reporting Enable (UNSPREQERREN):** Unsupported Request Error Reporting Enable
13	0h RO	**Poisoned TLP Error Reporting Enable (PTLPERREN):** Poisoned TLP Error Reporting Enable
12:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Interrupt Disable (INTDIS):** Interrupt Disable: Controls the ability of the function to generate INTx interrupts. 0: INTx allowed 1: INTx disabled

Bit Range	Default & Access	Field Name (ID): Description
9:6	0h RO	**Reserved (RSVD_2):** Reserved
5	0h RO	**Max Aligned Payload Size (MXAPAYLDSZ):** Max Aligned Payload Size - Reserved
4	0h RO	**Max Aligned Read Request Size (MXARDREQSZ):** Max Aligned Read Request Size - Reserved
3	0h RO	**Special Cycle Enable (SCEN):** Special Cycle Enable - Reserved
2	0h RW	**Bus Master Enable (BME):** Bus Master Enable: 0: Disable (default) 1: Enabled. Device may generate bus master transactions depending on its mode of operation.
1	0h RW	**Memory Space Enable (MSE):** Memory Space Enable Controls the GMM devices response to memory space accesses 0: Disabled (default) 1: Enabled. Device will respond to memory space accesses.
0	0h RO	**IO Space Enable (IOSE):** IO Space Enable. Not implemented.

6.3 Device Status (DSTS) — Offset 6h

The Status register to record status information for PCI/IOSF related events

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:0, F:3] + 6h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**Detected Parity Error (DPE):** This bit is Set by a Function whenever it receives a Poisoned TLP, regardless of the state the Parity Error Response bit in the Command register. On a Function with a Type 1 Configuration header, the bit is Set when the Poisoned TLP is received by its Primary Side. Note : some implementations use this error type as non-fatal error indication This bit is typically RWC. Change to RO as this bit is not in use
14	0h RO	**Signaled System Error (SSE):** This bit is Set when a Function sends an ERR_FATAL or ERR_NONFATAL Message, and the SERR# Enable bit in the Command register is 1. Note: some implementations use this error for fatal. When received all operations are aborted. This bit is typically RWC. Change to RO as this bit is not in use
13	0h RW/1C/V	**Received Master Abort (RMA):** This bit is Set when a Requester receives a Completion with Unsupported Request Completion Status. On a Function with a Type 1 Configuration header, the bit is Set when the Unsupported Request is received by its Primary Side.
12	0h RW/1C/V	**Received Target Abort (RTA):** This bit is set when a transaction abort is received to a GMM initiated transaction
11	0h RW/1C/V	**Signaled Target Abort (STA):** This bit is Set when a Function completes a Posted or Non-Posted Request as a Completer Abort error. This applies to a Function with a Type 1 Configuration header when the Completer Abort was generated by its Primary Side.
10:8	0h RO	**Reserved (RSVD):** Reserved
7	0h RO	**Fast Back-to-Back (FB2B):** Fast Back-to_Back (ignored by SW)
6:5	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
4	0h RO	**Capability List (CLIST):** Capability List 0 : no capability list 1 : the GMM contains a linked list of capabilities which is accessed via the CAPPTR register at offset 34h
3	0h RO/V	**Interrupt Status (INTSTS):** Interrupt Status Reflects the state of the interrupt in the device. Only when the Interrupt Disable bit in the command register is a 0 and this Interrupt Status bit is a 1, will this device send a virtual INTA. Setting the Interrupt Disable bit to a 1 has no effect on the state of this bit. This bit is controlled by HW. 0 : No interrupt pending 1 : Interrupt pending
2:0	0h RO	**Reserved (RSVD_2):** Reserved

6.4 RID: Revision ID DLCO: Class Code (RID_DLCO) — Offset 8h

RID: DLCO: This register identify the type of device. The values are as defined in PCI 3.0 bus specification in Appendix D. The GMM is identified as an Other system Peripheral

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:3] + 8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Base Class Code (BCC):** Base Class (Generic system Peripherals)
23:16	0h RO	**Sub Class Code (SCC):** Sub Class
15:8	0h RO	**Peripheral Interface (PROGINTERFACE):** Interface (other system peripheral)
7:0	0h RO/V	**Revision ID (RID):** Indicates stepping of this device. This register is set by side-band. All RID registers are sourced from a fuse/settings incremented for each stepping.

6.5 Cache Line Size (CLS) — Offset Ch

The system cacheline size in units of DWORDS

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:0, F:3] + Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Cache Line Size (CLS):** Implemented by PCI Express devices as a read-write field for legacy compatibility purposes but has no impact on any PCI Express device functionality

6.6 Header Type (HTYPE) — Offset Eh

This byte identifies the layout of the second part of the predefined header and whether or not the device contains multiple functions (GMM is a single-function device of basic configuration space format, so this register is Read-Only and hardwired to 0)

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:0, F:3] + Eh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**Multi Function Device (MFD):** Hardwired to 0 indicating this device is not a multi-function device.
6:0	0h RO	**Header Type (HT):** The value 00h, indicates a basic (i.e., single function) configuration space format.

6.7 Built-in Self Test (BIST) — Offset Fh

This register describes the BIST capability of GMM and since GMM doesnt support BIST, the register is configured as Read Only

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:0, F:3] + Fh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**BIST Capable (BISTCAP):** BIST Capable. Hardwired to 0 since this device does not implement BIST.
6	0h RO	**Start BIST (BISTST):** Start BIST. Hardwired to 0 since this device does not implement BIST
5:4	0h RO	**Reserved (RSVD):** Reserved
3:0	0h RO	**BIST Completion Code (BISTCC):** Hardwired to 0 since this device does not implement BIST.

6.8 GNA Base Address Low (GNABAL) — Offset 10h

GNA Base Address Low: Lower 32-bits of the GNA Base Address register. The GMM Base Address register may be accessed with Double Word (32bit) read/write operations. In 32-bit OS, the address specified may be limited by 32-bit of space, and the renaming bits must stay with their default values

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:3] + 10h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**Memory Base Address Low (BAL):** Memory Base Address Low Base address of this device's memory mapped IO space. A page of 4KB of address is used
11:4	0h RO	**Address Mask (ADDRMSK):** Address Mask Hardwired to 0s to indicate at least 4KB address range
3	0h RO	**PREF:** Hardwired to 0 indicating that this range is not prefetchable
2:1	0h RO	**Memory Type (MEMTY):** Memory Type: 00: 32 bit base address 01: reserved 10: 64-bit base address 11: reserved
0	0h RO	**Space Type (SPTY):** Space Type: Memory/IO Space Hardwired to 0 indicating that this is a Memory BAR

6.9 GNA Base Address High (GNABAH) — Offset 14h

GNA Base Address High: Upper 32-bits of the GNA Base Address register. The GNA Base Address register may be accessed with Double Word (32bit) read/write operations. In 32-bit OS, the address specified may be limited by 32-bit of space, and the renaming bits must stay with their default values

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:3] + 14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RW	**Memory Base Address High (Reserved) (BAR):** These bits must be loaded with zeros
6:0	0h RW	**Memory Base Address High (BAH):** Memory Base Address High - bits Includes the high bits of the base address used by 64-bit OS. Must hold zero for 32-bit OS

6.10 Sub System Vendor Identifiers (SSVI) — Offset 2Ch

This register is initialized to logic 0 by the assertion of reset. This register can be written only once after reset de-assertion it is locked for writes after that.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:0, F:3] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW/O	**Subsystem Vendor ID (SSVID):** Subsystem Vendor ID (SSVID): This is written by BIOS. No hardware action taken on this value.

6.11 Sub System Identifiers (SSI) — Offset 2Eh

This register is initialized to logic 0 by the assertion of reset. This register can be written only once after reset de-assertion it is locked for writes after that.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:0, F:3] + 2Eh	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW/O	**Subsystem ID (SSID):** Subsystem ID (SSID): This is written by BIOS. No hardware action taken on this value.

6.12 Capabilities Pointers (CAPP) — Offset 34h

This register gives MSI capability pointer offset

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:3] + 34h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD):** Reserved
7:0	0h RO	**Capability Pointer (CAPP):** Indicates that the MSI capability pointer offset is offset 90h

6.13 Interrupt Line (INTL) — Offset 3Ch

This register contains interrupt line routing information. The device itself does not use this value, rather it is used by device drivers and operating systems to determine priority and vector information.

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:0, F:3] + 3Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Interrupt Connection (INTCON):** Interrupt Connection Communicate interrupt line routing information. BIOS Requirement: POST software writes the routing information into this register as it initializes and configures the system. The value indicates to which input of the system interrupt controller this device's interrupt pin is connected

6.14 Interrupt Pin Register (INTP) — Offset 3Dh

tells which PCI legacy interrupt pin a device will use (GMM uses only IntA).

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:0, F:3] + 3Dh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:3	0h RO	**Reserved (RSVD):** Reserved
2:0	0h RO	**Legacy Interrupt (LEGINT):** When Legacy interrupts are used, function use legacy interrupt INTA.

6.15 Min Grant And Min Latency Register (MINGNTLAT) — Offset 3Eh

specifies a device's desired settings for Latency Timer values

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:0, F:3] + 3Eh	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	**Min Latency (MINLAT):** Reserved: Min Latency
7:0	0h RO	**Min Grant (MINGNT):** Reserved: Min Grant

6.16 Override Configuration Control (OVRCFGCTL) — Offset 40h

This register holds bits that may be used internal mechanisms in the GMM during debug operations. Special notes will be made to BIOS writers, if any 5 of these bits will need to be set to value other than default.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:3] + 40h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**RSVD (RSVD1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW	**Scoring Underrun/Overrun Error Check Disable (SUOECD):** Disable the consistency checks in GMM setup (sw calculated values GMMSCRLEN & GMMTELST) that are based on too-few or too-many elements in the DMA setup. Scoring Under-run is a scenario where DMA is out of data for scoring to proceed. 0 : Error Check is enabled,. 1 : Error Check is disabled.
15	0h RW	**PM_REQ (Package C state transition) NACK before Drain Enable (CXNBDEN):** 0: NACK PM_REQ is sent only after all outstanding completion is returned, and after master cycles are blocked. 1: NACK PM_REQ is sent before waiting for return of any outstanding completion
14	0h RW	**FLR_NP_Disable (FLRNPDIS):** 0: FLR operates normally.
13	0h RW	**FLR_Disable (FLRDIS):** 0: FLR operates normally. 1: FLR is disabled. Writes to start FLR will be ignored. FLR data structure is reported as usual.
12:10	0h RW	**RSVDCTL:** Reserved
9	0h RW	**PGCB Clock Trunk Clock Gating Enable (PGCBCGEN):** This bit, when set, allows the PGCB interface clock for GMM from SoC (pgcb_clk) to be gated when conditions are met by de-asserting GMMs clock request (pgcb_clkreq). When clear, GMM will always assert its clock request.
8	0h RW	**Sideband Clock Gating Enable (SBDCGEN):** This bit, when set, enables the sideband interface clock used for GMM bus interface operations (gated_side_clk) to be gated when conditions are met. When clear, clock gating is disabled.
7	0h RW	**Sideband Clock Trunk Clock Gating Enable (SBTCGEN):** This bit, when set, allows the sideband interface clock for GMM from SoC (iosfsf_clk) to be gated when conditions are met by deasserting GMMs clock request (iosfst_side_clkreq). When clear, GMM will always assert its clock request.
6	0h RW	**Sideband Clock Partition Clock Gating Enable (SBPCGEN):** This bit, when set, enables the IOSF sideband interface clock for GMM (side_clk) to be gated when conditions are met. When clear, clock gating is disabled
5	0h RW	**GMM Core Clock Gating Enable (GM_BB_GMMC_DCGEN):** This bit, when set, enables the primary interface clock used for GMM core operations (ggm_core_clk) to be gated when conditions are met. When clear, clock gating is disabled.
4	0h RW	**DMA engine Clock Gating Enable (GM_BB_DMA_DCGEN):** This bit, when set, enables the primary interface clock used for GMM DMA and MMU operations (ggm_bb_clk_ra) to be gated when conditions are met. When clear, clock gating is disabled.
3	0h RW	**Register Access Clock Gating Enable (GM_BB_RA_DCGEN):** This bit, when set, enables the primary interface clock used for GMM register access operations (ggm_bb_clk_ra) to be gated when conditions are met. When clear, clock gating is disabled.
2	0h RW	**Host interface Clock Gating Enable (GM_BB_HOST_DCGEN):** This bit, when set, enables the primary interface clock used for GMM bus interface operations (ggm_bb_clk_host) to be gated when conditions are met. When clear, clock gating is disabled
1	0h RW	**Partition Clock Gating Enable (PCGEN):** This bit, when set, enables the primary interface clock for GMM (gm_bb_clk) to be gated when conditions are met. When clear, clock gating is disabled.
0	0h RW	**Trunk Clock Gating Enable (TCGEN):** This bit, when set, allows the primary interface clock for GMM from SoC (iosfpf_clk) to be gated when conditions are met by deasserting GMMs clock request (iosfpt_prim_clkreq). When clear, GMM will always assert its clock request.

6.17 Message Signaled Interrupt Capability ID (MSICAPID) — Offset 90h

This register contains a pointer to the next item in the capabilities list which is the Power Management Capability and also helps to identify linked list item (capability structure) as being for MSI registers.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:0, F:3] + 90h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	**Pointer to Next Capability (NXTPTR):** Pointer to Next Capability This contains a pointer to the next item in the capabilities list which is the Power Management Capability
7:0	0h RO	**Capability ID (CAPID):** Capability ID Value of 05h identifies this linked list item (capability structure) as being for MSI registers.

6.18 Message Signaled Interrupt Message Control (MC) — Offset 92h

This register is defined to meet PCI Local Bus Specification 3.0 definition of MSI messages.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:0, F:3] + 92h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:9	0h RO	**Reserved (RSVD):** Reserved
8	0h RO	**Per-Vector Masking Capable (PVMCAP):** Per-Vector Masking Capable. 0- not supported by GMM
7	0h RO	**64-bit Address Capable (ADDR64CAP):** 64-bit Address Capable Hardwired to 0 to indicate that the function does not implement the upper 32 bits of the Message Address register and is incapable of generating a 64-bit memory address. This may need to change in future implementations when addressable system memory exceeds the 32bit/4GB limit.
6:4	0h RW	**Multiple Message Enable (MMEN):** Multiple Message Enable System software program this field to indicate the number of vectors allocated to the GMM. At least one vector must be allocated when the MSI interrupts are enabled. This value is ignored by HW as only a single vector is in use by GMM.
3:1	0h RO	**Multiple Message Capable (MMCAP):** Indicates to SW the number of vectors that the GMM module is requesting for use Value Number of Messages requested 000 1 001 2 (reserved) 010 4 (reserved) 011 8 (reserved) 100 16(reserved) 101 32(reserved) Other reserved
0	0h RW	**MSI Enable (MSIEN):** MSI Enable Controls the ability of GMM to generate MSI Messages. A device driver is prohibited from writing this bit to mask a functions service request. 0: MSI will not be generated 1: MSI will be generated. INTA will not be generated and INTA status is not set.

6.19 Message Signaled Interrupt Message Address (MA) — Offset 94h

This register is defined to meet PCI Local Bus Specification 3.0 definition of MSI messages

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:3] + 94h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**Message Address (MADDR):** Message Address Used by system software to assign an MSI address to the device. The device handles an MSI by writing the padded contents of the MD register to this address
1:0	0h RO	**Reserved (RSVD):** Reserved

6.20 Message Signaled Interrupt Message Data (MD) — Offset 98h

This register is defined to meet PCI Local Bus Specification 3.0 definition of MSI messages

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:3] + 98h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:0	0h RW	**Message Data (MDAT):** Message Data Base message data pattern assigned by system software and used to handle an MSI from the device. When the device must generate an interrupt request, it writes a 32-bit value to the memory address specified in the MA register. The upper 16 bits are always set to 0. The lower 16 bits are supplied by this register.

6.21 D0i3 Capability ID (D0I3CAPID) — Offset A0h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:0, F:3] + A0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	**Pointer to Next Capability (NXTPTR):** This contains a pointer to the next item in the capabilities list which is the Power Management Capability
7:0	0h RO	**Capability ID (CAPID):** Value of 09h identifies this linked list item (capability structure) is a vendor specific capability.

6.22 D0i3 Capability (D0I3CAP) — Offset A2h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:0, F:3] + A2h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:12	0h RO	**Vendor-Specific Capability ID (VSID):** Indicates that this Vendor Specific Capability is an Extended Capability, which use a VSEC 16-bit Externded VCapaiblity in the subsequent 4B., differentiating this from other vendor specific capabilities.
11:8	0h RO	**Vendor Specific Capability Revision (VSREV):** Reserved for VSID of Fh
7:0	0h RO	**Vendor Specific Capability Length (VSLEN):** This field indicates the number of bytes in this capability including the CapID and Cap registers.

6.23 D0i3 Vendor Extended Capability Register (D0I3VSEC) — Offset A4h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:3] + A4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RO	**Vendor Specific Extended Capability Length (VSECLEN):** Indicates that this Vendor Specific Capability is an Extended Capability, which use a VSEC 16-bit Externded VCapaiblity in the subsequent 4B., differentiating this from other vendor specific capabilities.
19:16	0h RO	**Vendor Specific Extended Capability Revision (VSREV):** For this revision of DevIdle, this field is 0h
15:0	0h RO	**Vendor Specific Extended Capability ID (VSECID):** DevIdle has been assigned the Intel VSEC ID of 10h

6.24 D0i3 SW LTR Pointer Register (D0I3SWLTRPTR) — Offset A8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:3] + A8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**SW LTR Update MMIO Offset Location (SWLTRLOC):** The value in this field is ignored as GMM does not support SW LTR
3:1	0h RO	**Base Address Register Number (BARNUM):** The value in this field is ignored as GMM does not support SW LTR
0	0h RO	**VALID:** Indicates the use of SW LTR by the function.GMM does not use SW LTR

6.25 D0i3 DevIdle Pointer Register (D0I3DEVIDLEPTR) — Offset ACh

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:0, F:3] + ACh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**DevIdle MMIO Offset Location (DEVIDLELOC):** This location pointer to the DevIdle register in MMIO space, as an offset from the BAR base.
3:1	0h RO	**Base Address Register Number (BARNUM):** The DevIdle is located in BAR0
0	0h RO	**VALID:** GMM has a DevIdle register

6.26 D0i3 DevIdle Power On Latency (D0I3DEVIDLEPOL) — Offset B0h

D0idle_5 Max_Power_On_Latency is set by BIOS at boot and read by device driver SW to calculate approximate cost of a D0idle entry + exit cycle. This allows driver to avoid idle entry in cases where device duty cycle is larger than D0idle entry + exit cycle.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:0, F:3] + B0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	**Reserved (RSVD):** Reserved
12:10	0h RO	**Power On Latency Scale (POLS):** Latency Scale multiplier: 010: 1us 011: 32us All other settings are reserved. This field is a RO as there is no need for BIOS programing of it.
9:0	0h RO	**Power On Latency Value (POLV):** A value of 0 indicates a power on latency of less than 1us. This field is a RO as there is no need for BIOS programing of it.

6.27 D0i3 Power Control Enables Register (PCE) — Offset B2h

This register controls the D0i3 features like Hardware Autonomous Enable,sleep enable, D3-Hot Enable, I3 Enable and PMC Request Enable

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:0, F:3] + B2h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:6	0h RO	**Reserved (RSVD):** Reserved
5	0h RW	**Hardware Autonomous Enable (HAE):** If set, then the IP may request a PG whenever it is idle. NOTE: If this bit is set, then bits[2:0] must be 000.
4	0h RO	**Reserved (RSVD_1):** Reserved
3	0h RO	**Sleep Enable (SE):** if clear, then IP will never asset Sleep to the retention flops. If set, then IP may assert Sleep during PGing. Note that some platforms may default this bit to 0, others to 1.
2	0h RW	**D3-Hot Enable (D3HE):** If set, then IP will PG when idle and the PMCSR[1:0] register in the IP =11.
1	0h RW	**I3 Enable (I3E):** If set, then IP will PG when idle and the D0i3 register (D0i3C[2] = 1) is set. NOTE: If bits [2:1] = 11, then the IP would PG whenever either PMCSR = 11 or the D0i3C.i3 bit is set.
0	0h RW	**PMC Request Enable (PMCRE):** If set, then IP will PG when idle and the PMC requests power gating by asserting the pmc_*_sw_pg_req_b signal.

6.28 Power Management Capability ID (PMCAPID) — Offset DCh

This register contains a pointer to next item in capabilities list and also helps to identify linked list item as being for PCI Power Management registers

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:0, F:3] + DCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	**Next Pointer (NXTPTR):** Next Pointer This contains a pointer to next item in capabilities list. This is the final capability in the list and must be set to 00h.
7:0	0h RO	**Capability Identifier (CAPID):** Capability Identifier Identifies this linked list item as being for PCI Power Management registers. This is compliant with the PCI Power Management Interface Specification.

6.29 Power Management Capability (PMCAP) — Offset DEh

This register describes the Power Management Capability of GMM

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:0, F:3] + DEh	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:11	0h RO	**PME Support (PMES):** PME Support This device does not support PMEB signal
10	0h RO	**D2 Support (D2S):** D2 This device does not support D2
9	0h RO	**D1 Support (D1S):** D1 This device does not support D1
8:6	0h RO	**Auxiliary Current (AUXC):** Auxiliary Current Reserved. Not applicable for GNA
5	0h RO	**Device Specific Initialization (DSI):** Device specific Initialization Indicates that this device requires device specific initialization before generic class device driver is to use it
4	0h RO	**Auxiliary Power (AUXP):** Aux Power This device does not use Aux power
3	0h RO	**PME Clock (PMEC):** PME Clock indicate this device does NOT support PMEB generation
2:0	0h RO	**VER:** Version Hardwired to 010b to indicate there are 4 bytes of power management registers implemented and that this device complies with revision 1.1 of the PCI Power Management Interface Specification.

6.30 Power Management Control Status (PMCS) — Offset E0h

This register has the status of PME Generation from D3(cold), Data Scale, Data Select, PME Enable and Power State

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:0, F:3] + E0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**PME Generation from D3 (cold) (PMEGD3):** PME Generation from D3 (cold) Not supported
14:13	0h RO	**Data Scale (DATSC):** Data Scale No support for Power Management Data register
12:9	0h RO	**Data Select (DATSEL):** Data Select No support for Power Management Data register
8	0h RO	**PME Enable (PMEE):** PME Enable PMEB is not supported
7:2	0h RO	**Reserved (RSVD):** Reserved
1:0	0h RW	**Power State (PS):** Indicates the current power state of this device and can be used to set the device into a new power state. If software attempts to write an unsupported state to this field, write operation must complete normally on the bus, but the data is discarded and no state change occurs. 00: D0 01: D1 (Not supported in this device.) 10: D2 (Not supported in this device.) 11: D3 Write of reserved values is ignored and state will not change. Support of D3cold does not require any special action. While in the D3hot state, this device can only act as the target of PCI configuration transactions (for power management control). This device also cannot generate interrupts or respond to MMR cycles in the D3 state. The device must return to the D0 state in order to be fully-functional.

6.31 FLR Capability ID (FLRCAPID) — Offset F0h

This register contains a pointer to next item in capabilities list and capablity of Advanced Features

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:0, F:3] + F0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	**Next Pointer (NXTPTR):** Next Pointer This contains a pointer to next item in capabilities list. This is the final capability in the list and must be set to 00h.
7:0	0h RO	**Capability Identifier (CAPID):** Capability Identifier A value that indicates FLR (Vendor specific value) 0 : 09h (FLR in use) A value of 09h in this register indicates that this is a FLR capabilities field.

6.32 FLR Capability Length And Version (FLRMISC) — Offset F2h

This register descibest the FLR Capability, TXP Capability and Capability Length

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:0, F:3] + F2h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:10	0h RO	**Reserved (RSVD):** Reserved
9	0h RO	**FLR Capability (FLRCAP):** Indicates support for Function Level Reset (FLR).
8	0h RO	**TXP Capability (TXPCAP):** Indicates that TP bit is supported
7:0	0h RO	**Capability Length (CAPLEN):** Capability Length This bit indicates the number of bytes this vendor specified capability requires. it has a value of 06h for the FLR capability

6.33 FLR Control Register (FLRCTL) — Offset F4h

This register controls the Functional Level reset operation of gmm

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:0, F:3] + F4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:1	0h RO	**Reserved (RSVD):** Reserved
0	0h WO	**Initiate FLR (INITFLR):** Writing 1 to this field starts the Functional Level Reset.This will act similar to the Abort + will bring all non-CFG registers to their reset value. The FLR is completed when the FLR status bit is cleared

6.34 FLR Status Register (FLRSTS) — Offset F5h

This register helps to identify whether FLR is in progress

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:0, F:3] + F5h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:1	0h RO	**Reserved (RSVD):** Reserved
0	0h RO/V	**Transaction Pending (XPEND):** Transaction Pending 0 : FLR not in progress 1 : FLR is in progress (due to internal operation or waiting for the completion of a non-posted transaction)

6.35 GMM Status Register (GNASTS)—Offset 80h

This register contains a quick status of the state of the GMM Scoring Accelerator.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved Always returns zero when read
23:18	0h RW/1C/V	**Reserved (RSVDSTS):** Reserved for future use
17	0h RW/1C/V	**Score has reached the saturation point (SAT):** Score has reached the saturation Internal math in score calculation has reached saturation (all 1 s) and thus is not accurate. Read 0 - Saturation didn t occurred 1 - Saturation occurred. Write : 0 - ignored 1 - clear the bit
16	0h RW/1C/V	**GMM output buffer is currently full. (FULL):** GMM output buffer is currently full. GMM scoring stall due to score output buffer being full and previous score-store DMA operation not completed yet. This is a backpressure situation that should not occur, the status flag is used to catch this condition. Read 0 - no back pressure occurred 1 - back pressure occurred Write : 0 - ignored 1 - clear the bit
15:9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO/V	**GMM Parameter out of Range (PARAMOORNG):** GMM Parameter out of Range 1 indicates an error casued by one of the GMM parameters being out of the operational range of the GMM Module. This bit is clear by the GMM Abort..
7	0h RO/V	**VA Out of Range Error (VAOORNG):** VA Out of Range Error 000 : no error n : DMA Channel n (1..5) has reached a VA out of range error. This field is clear by the GMM Abort..
6	0h RO/V	**PCIe Error Unexpected Completion (PCIEERRUNEXPCPL):** PCIe Error un Expected Completion 1 indicates an error that is related to the PCIe operation. The error is due to unexpected completion. This bit is clear by the GMM Abort..
5	0h RO/V	**PCIe Error DMA Req (PCIEERRDMA):** PCIe Error DMA Req 1 indicates an error that is related to the PCIe operation. The error is due to DMA. This bit is clear by the GMM Abort..
4	0h RO/V	**PCIe Error MMU Req (PCIEERRMMU):** PCIe Error MMU Req 1 indicates an error that is related to the PCIe operation. The error is due to MMU req. This bit is clear by the GMM Abort..
3	0h RO/V	**Compute Statistics Valid (COMPVALID):** Compute Statistics Valid 1 indicates the GMM collected performance statistics and that the value of stall and total compute cycles are valid. This bit is clear by the GMM Abort..

Bit Range	Default & Access	Field Name (ID): Description
2	0h RO/V	**Suspended due to Pause (SUSPPAUSE):** Suspended due to Pause 1 indicates that a GMM Scoring operation was suspended due to a SW write of 1 to the Pause bit. GMM execution is now suspended to allow software observation of current state. The bit is cleared when the operation is resumed or aborted.
1	0h RO/V	**Suspended Break Point Match (SUSPBREAKPT):** Suspended Break Point Match 1 indicates that a breakpoint match was reached and that GMM execution is now suspended to allow software observation of current state. The bit is cleared when the operation is resumed or aborted.
0	0h RO/V	**Scoring completed. (SCRCPL):** Scoring completed. Indicate that the scoring of the frame is completed but data may still be DMAed to memory. The bit is cleared when the GMM Abort is set.

6.36 GMM Control Register (GNACTRL)—Offset 84h

This register is used to initiation GMM Scoring Accelerator calculations.

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved Always returns zero when read
23:19	0h RW	**Reserved (RSVDCTL):** Reserved for future use
18	0h RW	**Power Management Quiet-Idle Disable (PMQID):** Power Management Quiet-Idle Disable 0 : module return to Idle immediately (by clearing Start bit) when Completion status bit is set. 1 : module remains in GMM scoring mode when until SW writes to the GMM Abort/clear bit a 1
17	0h RW	**PM Override Force Clock On (PMCLKON):** PM Override Force Clock On 1 : force the clocks to all GMM modules to be always on. Setting this bit impact the global power management of the device
16	0h RW	**PM Override Power On (PMPWRON):** PM Override Power On 1 : force power to the module to be always on. Setting this bit impact the global power management of the device

Bit Range	Default & Access	Field Name (ID): Description
15:12	0h RW	**Compute Statistics Enable (COMPSTATEN):** Compute Statistics Enable This fields control the enabling of the GMM Scoring Acceleration performance counters. When performance counting is enabled, the total scoring cycles counter is always on. In addition one of several reasons for stall may be measured to allow identifying the bottlenecks in the scoring operation. Bits Encoding 000 Performance counting is disabled 001 Count Total stall cycle 010 Wait for DMA Completion 011 Wait for MMU Transaltion Others reserved
11	0h RO	**Reserved (RSVD_1):** Reserved Always returns zero when read
10	0h RW	**Error Interrupt Enable (ERRINTEN):** Error Interrupt Enable 0 : Interrupt is not sent when the GMM execution is stopped due to an error. 1 : Interrupt is sent when execution is stopped due to an error as indicated by the setting of one of the Error status bits
9	0h RW	**Break Point Pause Interrupt Enable (BRKPTINTEN):** Break Point Pause Interrupt Enable : 0 : No interrupt sent on execution Pause due to break point. 1 : Interrupt is sent when execution is paused due to a reaching a break point.
8	0h RW	**Completion Interrupt Enable (CPLINTEN):** Completion Interrupt Enable 0 : Interrupt is not sent when the GMM execution is completed. 1 : Interrupt is sent when execution is completed and the Completion Status is set.
7	0h RO	**Reserved (RSVD_2):** Reserved
6:5	0h RW	**GNA Operation Mode (MODE):** 00: GMM; 01: XNN; Other: Reserved
4	0h RW/1S	**Resume Execution (RESUME):** Resume Execution 0 : ignored 1 : Resume execution when in the Pause states (status bit is set), due to Break point or a write to the Suspend bit. Pause status is cleared when execution resumes.
3	0h RW/1S	**Pause Execution (PAUSE):** Pause Execution Write behavior : 0 - ignored 1 - halts the execution of the GMM module at the next stop point. Once stopped the appropriate status bit indicates the reason of the pause.
2	0h RW/1S	**Abort Accelerator (ABORT):** Abort Accelerator Write behavior : 0 - ignored 1 - performs an abort operation. When written a 1, when the GMM is running or when halted due to an error or a break point. It will reset the GMM to its Idle condition and will clear any pending error bit in the MMIO registers. When the reset is completed the Start bit and all status bits are cleared. This bit retunes returns a zero when read.
1	0h RW	**Active List Enable (ALE):** Enables the use of the active state list. "1"- once enabled, the value in the active state list parameter registers are valid "0"-disabled

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW/1S	**Start GMM Accelerator (START):** Start GMM Accelerator Write behavior : 0 - ignored 1 - start execution of the GMM hardware based on the parameters set in the various register. When read the bit indicates the current status of the GMM execution. 0 - GMM is Idle 1 - GMM is executing. The Start bit is cleared by HW when GMM execution is completed (note that data might be in flight to memory, and pushed by status read completion or MSI). In case execution is halted due to error conditions, or due to a break point, the bit stay set, and only the halt status is flagged. The bit is also cleared when GMM operations is Aborted by a write of 1 to the respective control bit. .

6.37 GMM Management And Control Register (GNAMCTL)—Offset 88h

This register specifies the System Agent Frequency requirement and max number of allowed outstanding memory transactions in gmm

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: Fh

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Reserved (RSVD):** Reserved
8	0h RW	**System Agent High Frequency Required (SAHFREQ):** System Agent High Frequency Required 0 - GMM can operate at SA low frequency or high freq. 1 - GMM is scoring a large model and a SA high freq. is required during GMM active mode. This bit must be by SW while the GMM is Idle to guarantee proper operation.
7:4	0h RO	**Reserved (RSVD_1):** Reserved
3:0	Fh RW	**Max Outstanding Transaction Control (MAXOTC):** Max Outstanding Transaction Control -1 The number of outstanding transactions that may be pending to memory at any given time. Smaller numbers may be enforced due to lack of system resources or override by uArch control.

6.38 GMM Performance Total Cycle (GNAPTC)—Offset 8Ch

Indicates the number of cycles that the GMM was in Score in Progress state since the last score enable was started

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**GMM Performance Total Cycle (GNAPTC):** Indicates the number of cycles that the GMM was in Score in Progress state since the last score enable was started. A value of all 1 indicates saturation of the counter.

6.39 GMM Performance Stall Cycles (GNAPSC)—Offset 90h

Indicates the number of stall cycles that the GMM had since the last Score in Progress state since the last score enable was started

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**GMM Performance Stall Cycles (GNAPSC):** Indicates the number of stall cycles that the GMM had since the last Score in Progress state since the last score enable was started. The type of stall is defined in the A value of all 1 indicates saturation of the counter.

6.40 GMM Internal State Index (GNAISI)—Offset 94h

For debug assistance, the GNAISI register identifies the internal signal states by indexing to a list of pre-defined observation points. The actual internal signal state values are routed to the GNAPISVL and GNAPISVH registers

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD):** Reserved
10:0	0h RW	**Index (IND):** Index that select the proper internal status for the GMM module. Signal list is defined in Table 1-3

6.41 GMM Internal State Value Low (GNAPISVL)—Offset 98h

The lower 32 bits of the GMM internal signal states as identified by the GNAISI register

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Low Status Bits (GNAPISVL):** Low Status bits Low Status bits. Meaning of field depends on status register selected in the Index field.

6.42 GMM Internal State Value High (GNAPISVH)—Offset 9Ch

The higher 32 bits of the GMM internal signal states as identified by the GNAISI register

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**High Status bits (GNAPISVH):** High Status bits High Status bits applicable for some data elements. Meaning of field depends on status register selected in the Index field.

6.43 GMM Break Point Setup Low (GNABPL)—Offset A0h

This register may be read and written as a 32 bit register when the GMM scoring is off or suspended. The operation of the break point will use either Active list or GMM state index or both (in which case match occurs when either match is reached).

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:13	0h RW	**GMM States (GMMST):** The index of GMM state at which the break point match. 1 to 262144: are valid and indicate GMM state at which to stop (counting starts at 1). 0 : ignore the GMM state index (stop will use active list index and mixture component).
12:0	0h RW	**Mixture Components Number (MCNUM):** Mixture Components Number The index of the break point mixture component index. 1 to 4096 : are valid and indicate stop after the mixture component of that number (counting starts at 1). 0 : means stopping before the first mixture component.

6.44 GMM Break Point Setup High (GNABPH)—Offset A4h

This register may be read and written as a 32 bit register when the GMM scoring is off or suspended. The operation of the break point will use either Active list or GMM state index or both (in which case match occurs when either match is reached).

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Reserved1 (RSVD1):** Reserved
30:19	0h RO	**Reserved (RSVD):** Reserved
18:0	0h RW	**Active Index (ACTIND):** Active Index The index of Active List state at which the break point match. 1 to 262144: are valid and indicate GMM Active List index at which to stop (counting starts at 1). 0 : ignore the Active List state index (stop will use GMM state index and mixture component).

6.45 GMM Internal State Value (GNAD0i3C)—Offset A8h

This register is used as part of the D0i3 enter flow only. When that mode is not enabled this register is ignored by HW and its bits behavior is not guranteed.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 8h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD1):** Reserved
7:4	0h RO	**Reserved (RSVD):** Reserved
3	1h RW/1C/V	**Restore Required (RESTREQ):** Indicates the validity of the MMIO space registers and the need to restore them. The state may have been lost due to a reset or A MDfull power lost. SW clears the bit by writing a 1. This bit will be set on initial power up. Read : 0 no restore is require 1 : set by HW to indicate that a restore of state is required due to power loss. Write : 0 : ignored 1 : clear the bit When set (by HW), SW must restore state to the IP.
2	0h RW	**D0i3_SW (D0i3):** SW sets this bit to 1 to move the IP into the D0i3 state. Writing this bit to 0 will return the IP to the fully active D0 state (D0i0).
1	0h RW	**Interrupt Request (INTREQ):** SW sets this bit to 1 to ask for an interrupt to be generated on completion of the command. SW must clear or set this on each write to this register. 0 : interrupt will not be generated 1 : interrupt will be generated on the transition of 1-> 0 of the Command In Progress bit.
0	0h RO/V	**Command-In-Progress (CMDIP):** This bit is set by HW when a transition into or out of D0i3 command is received (a change 0->1 or 1 ->0 on D0i3 bit). The bit is changed to 0 when the enter/exit of D0i3 is completed. While set, the other bits in this register are not valid and it is illegal for SW to write to any bit in this register. When clear all the other bits in the register are valid and SW may write to any bit. If Interrupt Request bit [1] was set for the current command, HW may clear this bit before the interrupt has been made visible to SW, since when SW actually handles a particular interrupt is not visible to the HW. SW writes to this bit have no effect.

6.46 GNA Descriptor Base Address Register (GNADESBASE)—Offset B0h

GNA Descriptor Base Address Register

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: FFFFFFFFh

Bit Range	Default & Access	Field Name (ID): Description
31:0	FFFFFFFFh RW	**Base Physical Offset (GNADESBASE):** The upper 32-bit of he directory page address. The directory structure is always at the beginning of a 4KB aligned block that starts at this address

6.47 GNA Information Register (GNAIBUFFS)—Offset B4h

GNA Information Register

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 18h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD):** Reserved
7:0	18h RO	**GNA Input buffer size (GNAIBUFFS):** GNA Input Buffer Size. The size of the input buffer in KB.

6.48 GNASAI1L (GNASAI1L)—Offset 100h

GNA access Policy for MMIO Group (RW access).

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: FFF0FFFFh

Bit Range	Default & Access	Field Name (ID): Description
31:0	FFF0FFFFh RO	**GNASAI1 (GNASAI1L):** Lower 32 bits of GNA access Policy for MMIO Group (RW access). This is a RO register

6.49 GNASAI1H (GNASAI1H)—Offset 104h

GNA access Policy for MMIO Group (RW access).

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: FFF0FFFFh

Bit Range	Default & Access	Field Name (ID): Description
31:0	FFF0FFFFh RO	**GNASAI1H (GNASAI1H):** Upper 32 bits of GNA access Policy for MMIO Group (RW access). This is a RO register

6.50 GNASAI2 (GNASAI2L)—Offset 108h

GNA access Policy for Private registers Group (RW access).

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: FFF0FFFFh

Bit Range	Default & Access	Field Name (ID): Description
31:0	FFF0FFFFh RO	**GNASAI2 (GNASAI2L):** Lower 32 bits of GNA access Policy for Private registers Group (RW access). This is a RO register

6.51 GNASAI2 (GNASAI2H)—Offset 10Ch

GNA access Policy for Private registers Group (RW access).

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: FFF0FFFFh

Bit Range	Default & Access	Field Name (ID): Description
31:0	FFF0FFFFh RO	**Base Physical Offset (GNASAI2H):** Upper 32 bits of GNA access Policy for Private registers Group (RW access). This is a RO register

6.52 GNASAI2 (GNASAIV)—Offset 110h

GNA access Policy for Private registers Group (RW access).

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD):** Reserved
7:0	0h RO	**GNA SAIV Setup (GNASAIV):** Outgoing SAI of GNA

§ §

7 SMBus Registers

This chapter documents the registers in Bus: 0, Device 31, Function 1.

7.1 Vendor ID (VID) — Offset 0h

Value 8086h indicates Intel as the vendor

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:31, F:1] + 0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RO	**Vendor ID (VID):** Value indicates Intel as the vendor

7.2 Device ID (DID) — Offset 2h

Indicates the Device Number assigned for SMBus Host Controller Device ID.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:31, F:1] + 2h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RO/V	**Device ID (DID):** Indicates the device number assigned by the SIG.

7.3 Command (CMD) — Offset 4h

Contains various bits such as IO Space Enable, Memory Space Enable, Bus Master Enable etc.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:31, F:1] + 4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:11	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW	**Interrupt Disable (INTD):** 1 = Disables SMBus to assert its PIRQB# signal. Defaults to 0.
9	0h RO	**Fast Back to Back Enable (FBE):** Reserved as 0. Read Only.
8	0h RW	**SERR# Enable (SERRE):** 1 = Enables SERR# generation
7	0h RO	**Wait Cycle Control (WCC):** Reserved as 0. Read Only.
6	0h RW	**Parity Error Response (PER):** 1 = Sets Detected Parity Error bit when parity error is detected
5	0h RO	**VGA Palette Snoop (VGAPS):** Reserved as 0. Read Only.
4	0h RO	**Postable Memory Write Enable (PMWE):** Reserved as 0. Read Only.
3	0h RO	**Special Cycle Enable (SCE):** Reserved as 0. Read Only.
2	0h RO	**Bus Master Enable (BME):** Reserved as 0. Read Only.
1	0h RW	**Memory Space Enable (MSE):** 1= Enables memory mapped config space.
0	0h RW	**I/O Space Enable (IOSE):** 1= enables access to the SM Bus I/O space registers as defined by the Base Address Register.

7.4 Device Status (DS) — Offset 6h

Contains various bits such as Interrupt Status, Capabilities List Indicator, Signaled System Error, Detected Parity Error etc.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:31, F:1] + 6h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**Detected Parity Error (DPE):** Set when the bridge detects a parity error on the internal backbone. This bit gets set even if CMD.PERE is not set.
14	0h RW/1C	**Signaled System Error (SSE):** Set when the SMBus signals a system error to the internal SERR# logic.
13	0h RO	**Received Master Abort (RMA):** Reserved as 0.
12	0h RO	**Received Target Abort (RTA):** Reserved as '0'.
11	0h RW/1C	**Signaled Target-Abort Status (STA):** Reserved as 0.

Bit Range	Default & Access	Field Name (ID): Description
10:9	0h RO	**DEVSEL# Timing Status (DEVT):** This 2-bit field defines the timing for DEVSEL# assertion. These read only bits indicate the SoC's DEVSEL# timing when performing a positive decode. Note: SoC generates DEVSEL# with medium time. Note: It is not clear if a PCI master can write to SMBus controller.
8	0h RO	**Data Parity Error Detected (DPED):** Reserved as 0.
7	0h RO	**Fast Back-to-Back Capable (FBC):** Reserved as '1'.
6	0h RO	**User Definable Features (UDF):** Reserved as 0.
5	0h RO	**66 MHz Capable (C_66M):** Reserved as 0.
4	0h RO	**Capabilities List Indicator (CLI):** Hardwired to 0 because there are no capability list structures in this function.
3	0h RO	**Interrupt Status (INTS):** This bit indicates that an interrupt is pending. It is independent from the state of the Interrupt Enable bit in the command register.
2:0	0h RO	**Reserved (RSVD):** Reserved

7.5 Revision ID (RID) — Offset 8h

Indicates the part revision. This will reset to 0 but is expected to be overridden by the SetID IOSF-SB message.

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:31, F:1] + 8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Revision ID (RID):** The value reported in this register depends on theglobal revision ID for the PCH.

7.6 Programming Interface (PI) — Offset 9h

SMBus Controller has no programming interface.

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:31, F:1] + 9h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Programming Interface (PI):** No programming interface defined.

7.7 Sub Class Code (SCC) — Offset Ah

Indicates this device is a SMBus serial controller.

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:31, F:1] + Ah	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Sub Class Code (SCC):** A value of 05h indicates that this device is a SM Bus serial controller.

7.8 Base Class Code (BCC) — Offset Bh

Indicates this device is a serial controller.

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:31, F:1] + Bh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Base Class Code (BCC):** A value of 0Ch indicates that this device is a serial controller

7.9 SMBus Memory Base Address_31_0 (SMBMBAR_31_0) — Offset 10h

Provides the 32 byte system memory base address for the SoC SMB logic.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:1] + 10h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**BA:** Provides the 32 byte system memory base address for the SoC SMB logic.
7:4	0h RO	**HARDWIRED_0:** Hardwired to 0.
3	0h RO	**PREF:** Hardwired to 0. Indicated that SMBMBAR is not pre- fetchable
2:1	0h RO	**Address Range (ADDRNG):** Indicates that this SMBMBAR can be located anywhere in 64 bit address space

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**Memory Space Indicator (MSI):** Indicates that the SMB logic is memory mapped.

7.10 SMBus Memory Base Address_63_32 (SMBMBAR_63_32) — Offset 14h

Bits 63-32 of SMbus Memory Base Address

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:1] + 14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**BA:** Bits 63-32 of SMbus Memory Base Address

7.11 SMB Base Address (SBA) — Offset 20h

Provides the 32 byte system I/O base address for the SMB logic.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:1] + 20h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:5	0h RW	**BA:** Provides the 32 byte t system I/O base address for the SMB logic.
4:1	0h RO	**Reserved (RSVD_1):** Reserved
0	0h RO	**IO Space Indicator (IOSI):** This read-only bit always is 1, indicating that the SMB logic is I/O mapped.

7.12 SVID — Offset 2Ch

BIOS sets the value in this register to identify the Subsystem Vendor ID. The SMBus SVID register, in combination with the SMBus Subsystem ID register, enables the operating system to distinguish each subsystem from the others.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:31, F:1] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW/O	**SVID:** BIOS sets the value in this register to identify the Subsystem Vendor ID. The SMBus SVID register, in combination with the SMBus Subsystem ID register, enables the operating system to distinguish each subsystem from the others. Note: The software can write to this register only once per core well reset. Writes should be done as a single 16-bit cycle.

7.13 SID — Offset 2Eh

BIOS can write to this register to identify the Subsystem ID. The SID register, in combination with the SVID, enable the operating system to distinguish each subsystem from other(s).

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:31, F:1] + 2Eh	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW/O	**SID:** BIOS can write to this register to identify the Subsystem ID. The SID register, in combination with the SVID, enable the operating system to distinguish each subsystem from other(s). Note: The software can write to this register only once per core well reset. Writes should be done as a single 16-bit cycle.

7.14 Interrupt Line (INTLN) — Offset 3Ch

This data is not used by the hardware. It is to communicate to software the interrupt line that the interrupt pin is connected to PIRQB#.

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:31, F:1] + 3Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Interrupt Line (INTLN):** This data is not used by the hardware. It is to communicate to software the interrupt line that the interrupt pin is connected to PIRQB#.

7.15 Interrupt Pin (INTPN) — Offset 3Dh

Defines the interrupt pin to be used by the SMBus controller.

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:31, F:1] + 3Dh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW/O	**Interrupt Pin (INTPN):** This defines the interrupt pin to be used by the SMBus controller. Bits : Pins 0h : No Interrupt 1h : INTA# 2h : INTB# 3h : INTC# 4h : INTD# 5h-Fh : Reserved.

7.16 Host Configuration (HCFG) — Offset 40h

Contains the HST_EN, SMB_SMI_EN, I2C_EN, SSRESET and SPD Write Disable bit.

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:31, F:1] + 40h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:5	0h RO	**Reserved (RSVD):** Reserved
4	0h RW/1L	**SPD Write Disable (SPDWD):** This bit is must be set to '1' to disable writes to SPD which are on Host SMB address ranges A0h - AEh. The SMBus range is unwriteable until next platform reset. HW Default is '0.' Note: This bit is RW O and will be reset on PLTRST# reset. This should be set by BIOS memory reference code to '1'. SW can only program this bit when both HCTL(6) = 0 (START) and HSTS(0) = 0 (HBSY), else it may result in an undefined behavior. [Noted in HSTS.HBSY that states "No SMB registers should be accessed while HSTS.HBSY bit is set, the same will apply to SPD write disable. 3 RW 0 SSRESET : Soft SMBUS Reset: When this bit is 1,
3	0h RW	**SSRESET:** Soft SMBUS Reset: When this bit is 1, the SMbus state machine and logic in PCH is reset. The HW will reset this bit to 0 when reset operation is completed.
2	0h RW	**I2C_EN (I2CEN):** When this bit is 1, the SoC is enabled to communicate with I2C devices. This will change the formatting of some commands. When this bit is 0, behavior is for SMBus.
1	0h RW	**SMB_SMI_EN (SSEN):** When this bit is set, any source of an SMB interrupt will instead be routed to generate an SMI#.
0	0h RW	**HST_EN (HSTEN):** When set, the SMB Host Controller interface is enabled to execute commands. The HST_INT_EN bit needs to be enabled in order for the SMB Host Controller to interrupt or SMI#. Additionally, the SMB Host Controller will not respond to any new requests until all interrupt requests have been cleared.

7.17 B-Unit Copy of the TCO Base Address Register for Legacy SMBUS (B_CR_TCOBASE_SMBUS) — Offset 50h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:1] + 50h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:5	0h RW/V	**TCO Base Address (TCOBA):** Provides the 32 bytes of I/O space for TCO logic, mappable anywhere in the 64k I/O space on 32-byte boundaries.
4:0	0h RO	**Reserved (RSVD):** Reserved

7.18 B-Unit Copy of the TCO Control Register for Legacy SMBUS (B_CR_TCOCTL_SMBUS) — Offset 54h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:1] + 54h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Reserved (RSVD):** Reserved
8	0h RW/V	**TCO Base Enable (TCO_BASE_EN):** When set, decode of the I/O range pointed to by the TCO base register is enabled.
7:0	0h RO	**Reserved (RSVD):** Reserved

7.19 Manufacturer's ID (MANID) — Offset F8h

This reflects the value of the Manufacturers ID which provides information on the details of the chip revision. Implementation Note: A single Manufacturers ID is implemented based on information in the Fuse block and distributed to the devices via a message. This register is assigned during the boot flow using the SetID message based on values found in the fuse block. All fields except MID will be reset by hardware to zero and set only by the SetID message.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:1] + F8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD):** Reserved
27:24	0h RO/V	**Dot Portion of Process ID (DPID):** Indicates the dot process as x.8
23:16	0h RO/V	**Stepping ID (SID):** This field is incremented for each stepping of the part. Note that this field can be used by software to differentiate steppings when the Revision ID may not change.

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO/V	**Manufacturer (MAFA):** 0Fh = Intel
7:0	0h RO/V	**Process/Dot (PD):** Indicates that the process base is 1271.

7.20 Host Status Register Address (HSTS)—Offset 0h

Contains the Host Status bits i.e. Host Busy, Interrupt, Device Error, Bus Error, Failed, SMBALERT_STS, In Use Status and Byte Done Status. All status bits are set by hardware and cleared by the software writing a one to the particular bit position. Writing a zero to any bit position has no affect.

Access Method

Type: MEM Register
(Size: 8 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW/1C	**BYTE_DONE_STS (BDS):** This bit will be set to 1 when the host controller has received a byte (for Block Read commands) or if it has completed transmission of a byte (for Block Write commands) when the 32-byte buffer is not being used. Note that this bit will be set, even on the last byte of the transfer. Software clears the bit by writing a 1 to the bit position. This bit is not set when transmission is due to the D110 interface heartbeat. This bit has no meaning for block transfers when the 32- byte buffer is enabled. Note: When the last byte of a block message is received, the host controller will set this bit. However, it will not immediately set the INTR bit (bit 1 in this register). When the interrupt handler clears the BYTE_DONE_STS bit, the message is considered complete, and the host controller will then set the INTR bit (and generate another interrupt). Thus, for a block message of n bytes, the SoCSoC will generate n+1 interrupts. The interrupt handler needs to be implemented to handle these cases.
6	0h RW/1C	**In Use Status (IUS):** After a full PCI reset, a read to this bit returns a 0. After the first read, subsequent reads will return a 1. A write of a 1 to this bit will reset the next read value to 0. Writing a 0 to this bit has no effect. Software can poll this bit until it reads a 0, and will then own the usage of the host controller. This bit has no other effect on the hardware, and is only used as semaphore among various independent software threads that may need to use the SoCs SMBus logic.
5	0h RW/1C	**SMBALERT_STS (SMSTS):** SoC sets this bit to a 1 to indicates source of the interrupt or SMI# was the SMBAlert# signal. Software resets this bit by writing a 1 to this location. This bit should also be cleared by RSMRST# (but not PLTRST#).

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW/1C	**Failed (FAIL):** When set, this indicates that the source of the interrupt or SMI# was a failed bus transaction. This is set in response to the KILL bit being set to terminate the host transaction.
3	0h RW/1C	**Bus Error (BERR):** When set, this indicates the source of the interrupt or SMI# was a transaction collision.
2	0h RW/1C	**Device Error (DERR):** When set, this indicates that the source of the interrupt or SMI# was due one of the following: Illegal Command Field Unclaimed Cycle (host initiated) Host Device Time-out Error. CRC Error
1	0h RW/1C	**Interrupt (INTR):** When set, this indicates that the source of the interrupt or SMI# was the successful completion of its last command.
0	0h RW/1C	**Host Busy (HBSY):** A 1 indicates that the SoC is running a command from the host interface. No SMB registers should be accessed while this bit is set. Exception: The BLOCK DATA REGISTER can be accessed when this bit is set ONLY when the SMB_CMD bits (in Host control register) are programmed for Block command or I2C Read command. This is necessary in order to check the DONE_STS bit.

7.21 Host Control Register (HCTL)—Offset 2h

Contains the interrupt generation bit, KILL, SMB CMD, Last Byte, Start and Packet Error Checking bits. Note: A read to this register will clear the pointer in the 32-byte buffer.

Access Method

Type: MEM Register (Size: 8 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**PEC_EN (PEC_EN):** When set to 1, this bit causes the host controller to perform the SMBus transaction with the Packet Error Checking phase appended. For writes, the value of the PEC byte is transferred from the PEC Register. For reads, the PEC byte is loaded in to the PEC Register. When this bit is cleared to 0, the SMBus host controller does not perform the transaction with the PEC phase appended. This bit must be written prior to the write in which the START bit is set.
6	0h RW	**START (START):** This write-only bit is used to initiate the command described in the SMB_CMD field. All registers should be setup prior to writing a 1 to this bit position. This bit always reads zero. The HOST_BUSY bit in the Host Status register (offset 00h) can be used to identify when the SoC has finished the command.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**LAST_BYTE (LAST_BYTE):** Used for I2C Read commands as an indication that the next byte will be the last one to be received for that block. The algorithm and usage model for this bit will be as follows (assume a message of n bytes): A. When the software sees the BYTE_DONE_STS bit set (bit 7 in the SMBus Host Status Register) for each of bytes 1 through n-2 of the message, the software should then read the Block Data Byte Register to get the byte that was just received. B. After reading each of bytes 1 to n-2 of the message, the software will then clear the BYTE_DONE_STS bit. C. After receiving byte n-1 of the message, the software will then set the LAST BYTE bit. The software will then clear the BYTE_DONE_STS bit. D. The SoC will then receive the last byte of the message (byte n). However, the SoC state machine will see the LAST BYTE bit set, and instead of sending an ACK after receiving the last byte, it will instead send a NAK. E. After receiving the last byte (byte n), the software will still clear the BYTE_DONE_STS bit. However, the LAST_BYTE bit will be irrelevant at that point. Note: This bit may be set when the TCO timer causes the SECOND_TO_STS bit to be set. See the TCO2_STS Register in Volume 1, bit 1 for more details on that bit. The SMBus device driver should clear the LAST_BYTE bit (if it is set) before starting any new command. Note: In addition to I2C Read Commands, the LAST_BYTE bit will also cause Block Read/Write cycles to stop prematurely (at the end of the next byte).

Bit Range	Default & Access	Field Name (ID): Description
4:2	0h RW	**SMB_CMD (SMB_CMD):** As shown by the bit encoding below, indicates which command the SoC is to perform. If enabled, the SoC will generate an interrupt or SMI# when the command has completed If the value is for a non-supported or reserved command, the SoC will set the device error (DEV_ERR) status bit and generate an interrupt when the START bit is set. The SoC will perform no command, and will not operate until DEV_ERR is cleared. Val - Command Description: 000 - Quick: The slave address and read/write value (bit 0) are stored in the tx slave address register. 001 - Byte: This command uses the transmit slave address and command registers. Bit 0 of the slave address register determines if this is a read or write command. 010 - Byte Data: This command uses the transmit slave address, command, and DATA0 registers. Bit 0 of the slave address register determines if this is a read or write command. If it is a read, the DATA0 register will contain the read data. 011 - Word Data: This command uses the transmit slave address, command, DATA0 and DATA1 registers. Bit 0 of the slave address register determines if this is a read or write command. If it is a read, after the command completes the DATA0 and DATA1 registers will contain the read data. 100 - Process Call: This command uses the transmit slave address, command, DATA0 and DATA1 registers. Bit 0 of the slave address register determines if this is a read or write command. After the command completes, the DATA0 and DATA1 registers will contain the read data. 101 - Block: This command uses the transmit slave address,command, and DATA0 registers, and the Block Data Byte register. For block write, the count is stored in the DATA0 register and indicates how many bytes of data will be transferred. For block reads, the count is received and stored in the DATA0 register. Bit 0 of the slave address register selects if this is a read or write command. For writes, data is retrieved from the first n (where n is equal to the specified count) addresses of the SRAM array. For reads, the data is stored in the Block Data Byte register. 110 - I2C Read: This command uses the transmit slave address, command, DATA0, DATA1 registers, and the Block Data Byte register. The read data is stored in the Block Data Byte register. The SoC will continue reading data until the NAK is received. 111 - Block-Process: This command uses the transmit slave address, command, DATA0 and the Block Data Byte register. For block write, the count is stored in the DATA0 register and indicates how many bytes of data will be transferred. For block read, the count is received and stored in the DATA0 register. Bit 0 of the slave address register always indicate a write command. For writes, data is retrieved from the first m (where m is equal to the specified count) addresses of the SRAM array. For reads, the data is stored in the Block Data Byte register. Note: E32B bit in the Auxiliary Control Register must be set for this command to work.
1	0h RW	**KILL (KILL):** When set, kills the current host transaction taking place, sets the FAILED status bit, and asserts the interrupt (or SMI#) selected by the SMB_INTRSEL field. This bit, once set, must be cleared to allow the SMB Host Controller to function normally.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**INTREN (INTREN):** Enable the generation of an interrupt or SMI# upon the completion of the command.

7.22 Host Command Register (HCMD)—Offset 3h

This eight bit field is transmitted by the host controller in the command field of the SMB protocol during the execution of any command.

Access Method

Type: MEM Register
(Size: 8 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Host Command Register (HCMD):** This eight bit field is transmitted by the host controller in the command field of the SMB protocol during the execution of any command.

7.23 Transmit Slave Address Register (TSA)—Offset 4h

This register is transmitted by the host controller in the slave address field of the SMB protocol. This is the address of the target.This eight bit field is transmitted by the host controller in the command field of the SMB protocol during the execution of any command.

Access Method

Type: MEM Register
(Size: 8 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:1	0h RW	**ADDRESS (ADDR):** 7-bit address of the targeted slave. Note: Writes to TSA values of A0h - AEh are blocked depending on the setting of the SPD write disable bit in HCFG - HostConfiguration.
0	0h RW	**RW (RW):** Direction of the host transfer. 1 = read, 0 = write

7.24 Data 0 Register (HD0)—Offset 5h

This register contains the eight bit data sent in the DATA0 field of the SMB protocol.

Access Method

Type: MEM Register
(Size: 8 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**DATA0/COUNT (DATA0_COUNT):** This field contains the eight bit data sent in the DATA0 field of the SMB protocol. For block write commands, this register reflects the number of bytes to transfer. This register should be programmed to a value between 1 and 32 for block counts. A count of 0 or a count above 32 will result in unpredictable behavior. The host controller does not check or log illegal block counts.

7.25 Data 1 Register (HD1)—Offset 6h

This eight bit register is transmitted in the DATA1 field of the SMB protocol during the execution of any command

Access Method

Type: MEM Register
(Size: 8 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**DATA1 (DATA1):** This eight bit register is transmitted in the DATA1 field of the SMB protocol during the execution of any command.

7.26 Host Block Data (HBD)—Offset 7h

This is a register containing a byte of data to be sent on a block write or read . Or a pointer into a 32- byte block array.

Access Method

Type: MEM Register
(Size: 8 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Block Data (BDTA):** This is either a register, or a pointer into a 32- byte block array, depending upon whether the E32B bit is set in the Auxiliary Control register. When the E32B bit is cleared, this is a register containing a byte of data to be sent on a block write or read from on a block read, just as it behaved on the SoC. When the E32B bit is set, reads and writes to this register are used to access the 32-byte block data storage array. An internal index pointer is used to address the array, which is reset to 0 by reading the HCTL register (offset 02h). The index pointer then increments automatically upon each access to this register. The transfer of block data into (read) or out of (write) this storage array during an SMBus transaction always starts at index address 0. When the E2B bit is set, for writes, software will write up to 32-bytes to this register as part of the setup for the command. After the Host Controller has sent the Address, Command, and Byte Count fields, it will send the bytes in the SRAM pointed to by this register. When the E2B bit is cleared for writes, software will place a single byte in this register. After the host controller has sent the address, command, and byte count fields, it will send the byte in this register. If there is more data to send, software will write the next series of bytes to the SRAM pointed to by this register and clear the DONE_STS bit. The controller will then send the next byte. During the time between the last byte being transmitted to the next byte being transmitted, the controller will insert wait-states on the interface. When the E2B bit is set for reads, after receiving the byte count into the Data0 register, the first series of data bytes go into the SRAM pointed to by this register. If the byte count has been exhausted or the 32-byte SRAM has been filled, the controller will generate an SMI# or interrupt (depending on configuration) and set the DONE_STS bit. Software will then read the data. During the time between when the last byte is read from the SRAM to when the DONE_STS bit is cleared, the controller will insert waitstates on the interface.

7.27 Packet Error Check Data Register (PEC)—Offset 8h

Note: This register may reside in either the core well or the suspend well. To simplify the implementation, this register will be in the suspend well with the suspend well version of PCI reset (URST33B). This register contains the 8-bit CRC value that is used as the Packet Error Check on SMBus. For writes, this register is written by software prior to running the command. For reads, this register is read by software after the read command is completed on SMBus.

Access Method

Type: MEM Register (Size: 8 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**PEC_DATA (PEC_DATA):** This 8-bit register is written with the SMBus PEC data prior to a write transaction. For read transactions, the PEC data is loaded from the SMBus into this register and is then read by software. Software must ensure that the INUSE_STS bit is properly maintained to avoid having this field over-written by a write transaction following a read transaction.

7.28 Receive Slave Address Register (RSA)—Offset 9h

Contains the slave address that the SoC decodes for read and write cycles.

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 44h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**Reserved (RSVD)**
6:0	44h RW	**SLAVE_ADDR[6:0] (SA_6_0):** This field is the slave address that the SoC decodes for read and write cycles. The default is not 0 so that it can respond even before the CPU comes up (or if the CPU is dead). This register is reset by RSMRST#, but not by PLTRST#.

7.29 Slave Data Register (SD)—Offset Ah

The 16-bit data value written by the external SMBus master.

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RO/V	**SLAVE_DATA[15:0] (SD_15_0):** This field is the 16-bit data value written by the external SMBus master. The CPU can then read the value from this register. This register is reset by RSMRST#, but not by PLTRST#. SLAVE_DATA[7:0] corresponds to the Data Message Byte 0 at Slave Write Register 4 in the table. SLAVE_[15:8] corresponds to the Data Message Byte 1 at Slave Write Register 5 in the table.

7.30 Auxiliary Status (AUXS)—Offset Ch

Contains the CRC Error, the SMT1EN, SMT2EN and SMT3EN bits. All bits in this register are in the core well.

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:5	0h RO	**Reserved (RSVD)**
4	0h RO/V	**SMT3EN (SMT3EN):** Register value determines if SMT3 is enabled and connected on SMLink1. Disabled means the device is not connected to the pins 0: Disable 1: Enable
3	0h RO/V	**SMT2EN (SMT2EN):** Register value determines if SMT2 is enabled and connected on SMLink0. Disabled means the device is not connected to the pins 0: Disable 1: Enable
2	0h RO/V	**SMT1EN (SMT1EN):** Register value determines if SMT1 is enabled and connected on SMBus. Disabled means the device is not connected to the pins 0: Disables 1: Enable
1	0h RO	**Reserved (RSVD_1)**
0	0h RW/1C	**CRC Error (CRCE):** This bit is set if a received message contained a CRC error. When this bit is set, the DERR bit of the host status register will also be set. This bit will be set by the controller if a software abort occurs in the middle of the CRC portion of the cycle or an abort happens after SoC has received the final data bit transmitted by external slave.

7.31 Auxiliary Control (AUXC)—Offset Dh

Contains the Automatically Append CRC and Enable 32-byte buffer bits. All bits in this register are in the resume well.

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:2	0h RO	**Reserved (RSVD)**

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Enable 32-byte Buffer (E32B):** When set, the Host Block Data register is a pointer into a 32-byte buffer, as opposed to a single register. This enables the block commands to transfer or receive up to 32-bytes before the SoC generates an interrupt.
0	0h RW	**Automatically Append CRC (AAC):** When set, the SoC will automatically append the CRC. This bit must not be changed during SM Bus transactions, or undetermined behavior will result. It should be programmed only once during the lifetime of the function.

7.32 SMLINK_PIN_CTL Register (SMLC)—Offset Eh

Contains the SMLINK[0]_CUR_STS, SMLINK[1]_CUR_STS and SMLINK_CLK_CTL bits. Note: This register is in the resume well and is reset by RSMRST#

Access Method

Type: MEM Register (Size: 8 bits)

Device:
Function:

Default: 4h

Bit Range	Default & Access	Field Name (ID): Description
7:3	0h RO	**Reserved (RSVD)**
2	1h RW	**SMLINK_CLK_CTL (SMLINK_CLK_CTL):** 0 = SoC will drive the SMLINK[0] pin low, independent of what the other SMLINK logic would otherwise indicate for the SMLINK(0) pin. 1 = The SMLINK[0] pin is Not overdriven low. The other SMLINK logic controls the state of the pin.
1	0h RO/V	**SMLINK[1]_CUR_STS (SMLINK1_CUR_STS):** This bit has a default value that is dependent on an external signal level. This returns the value on the SMLINK[1] pin. It will be 1 to indicate high, 0 to indicate low. This allows software to read the current state of the pin.
0	0h RO/V	**SMLINK[0]_CUR_STS (SMLINK0_CUR_STS):** This bit has a default value that is dependent on an external signal level. This returns the value on the SMLINK[0] pin. It will be 1 to indicate high, 0 to indicate low. This allows software to read the current state of the pin.

7.33 SMBUS_PIN_CTL Register (SMBC)—Offset Fh

Contains the SMBCLK_CUR_STS, SMBDATA_CUR_STS and SMBCLK_CTL bits. Note: This register is in the resume well and is reset by RSMRST#

Access Method

Type: MEM Register
(Size: 8 bits)

Device:
Function:

Default: 4h

Bit Range	Default & Access	Field Name (ID): Description
7:3	0h RO	**Reserved (RSVD)**
2	1h RW	**SMBCLK_CTL (SMBCLK_CTL):** 0 = SoC will drive the SMBCLK pin low, independent of what the other SMB logic would otherwise indicate for the SMBCLK pin. 1 = The SMBCLK pin is Not overdriven low. The other SMBus logic controls the state of the pin.
1	0h RO/V	**SMBDATA_CUR_STS (SMBDATA_CUR_STS):** This bit has a default value that is dependent on an external signal level. This returns the value on the SMBDATA pin. It will be 1 to indicate high, 0 to indicate low. This allows software to read the current state of the pin.
0	0h RO/V	**SMBCLK_CUR_STS (SMBCLK_CUR_STS):** This bit has a default value that is dependent on an external signal level. This returns the value on the SMBCLK pin. It will be 1 to indicate high, 0 to indicate low. This allows software to read the current state of the pin.

7.34 Slave Status Register (SSTS)—Offset 10h

This bit is set to 1 when it has completely received a successful Host Notify Command on the SMBus pins. All bits in this register are implemented in the 64 kHz clock domain. Therefore, software must poll the register until a write takes effect before assuming that a write has completed internally.

Access Method

Type: MEM Register
(Size: 8 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:1	0h RO	**Reserved (RSVD)**

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW/1C	**HOST_NOTIFY_STS (HNS):** The SoC sets this bit to a 1 when it has completely received a successful Host Notify Command on the SMBus pins. Software reads this bit to determine that the source of the interrupt or SMI# was the reception of the Host Notify Command. Software clears this bit after reading any information needed from the Notify address and data registers by writing a 1 to this bit. Note that the SoC will allow the Notify Address and Data registers to be over-written once this bit has been cleared. When this bit is 1, the SoC will NACK the first byte (host address) of any new Host Notify commands on the SMBus. Writing a 0 to this bit has no effect.

7.35 Slave Command Register (SCMD)—Offset 11h

Contains the HOST_NOTIFY_INTREN, HOST_NOTIFY_WKEN and SMBALERT_DIS bits. All bits in this register are implemented in the 64 kHz clock domain. Therefore, software must poll the register until a write takes effect before assuming that a write has completed internally. Also, software must confirm the prior written value before writing to the register again.

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:3	0h RO	**Reserved (RSVD)**
2	0h RW	**SMBALERT_DIS (SMB_D):** Software sets this bit to 1 to block the generation of the interrupt or SMI# due to the SMBALERT# source. This bit is logically inverted and ANDed with the SMBALERT_STS bit. The resulting signal is distributed to the SMI# and/or interrupt generation logic. This bit does not effect the wake logic.
1	0h RW	**HOST_NOTIFY_WKEN (HNW):** Software sets this bit to 1 to enable the reception of a Host Notify command as a wake event. When enabled this event is ORed in with the other SMBus wake events and is reflected in the SMB_WAK_STS bit of the General Purpose Event 0 Status register.
0	0h RW	**HOST_NOTIFY_INTREN (HNI):** Software sets this bit to 1 to enable the generation of interrupt or SMI# when HOST_NOTIFY_STS is 1. This enable does not affect the setting of the HOST_NOTIFY_STS bit. When the interrupt is generated, either PIRQB or SMI# is generated, depending on the value of the SMB_SMI_EN bit (D31, F3, Off40h, B1). If the HOST_NOTIFY_STS bit is set when this bit is written to a 1, then the interrupt (or SMI#) will be generated. The interrupt (or SMI#) is logically generated by ANDing the STS and INTREN bits.

7.36 Notify Device Address Register (NDA)—Offset 14h

Contains the 7-bit device address received during the Host Notify protocol of the SMBUS 2.0 specification.

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:1	0h RO/V	**DEVICE_ADDRESS (Dev_Addr):** This field contains the 7-bit device address received during the Host Notify protocol of the SMBus 2.0 specification. Software should only consider this field valid when the HOST_NOTIFY_STS bit is set to 1.
0	0h RO	**Reserved (RSVD)**

7.37 Notify Data Low Byte Register (NDLB)—Offset 16h

Contains the low byte of data received during the Host Notify protocol of the SMBUS 2.0 specification.

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**DATA_LOW_BYTE (DLB):** This field contains the first (low) byte of data received during the Host Notify protocol of the SMBus 2.0 specification. Software should only consider this field valid when the HOST_NOTIFY_STS bit is set to 1.

7.38 Notify Data High Byte Register (NDHB)—Offset 17h

Contains the high byte of data received during the Host Notify protocol of the SMBUS 2.0 specification.

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**DATA_HIGH_BYTE (DHB):** This field contains the second (high) byte of data received during the Host Notify protocol of the SMBus 2.0 specification. Software should only consider this field valid when the HOST_NOTIFY_STS bit is set to 1.

7.39 TCO_RLD Register (TRLD)—Offset 0h

The values in this register have no effects on the TCO Timer hardware.

Access Method

Type: IO Register
(Size: 16 bits)

Device:
Function:

Default: 4h

Bit Range	Default & Access	Field Name (ID): Description
15:10	0h RO	**Reserved (RSVD)**
9:0	4h RW	**TCORLD (TCORLD):** Reading this register will return the current count of the TCO timer. Writing any value to this register will reload the timer to prevent the timeout.

7.40 TCO_DAT_IN Register (TDI)—Offset 2h

Data Register for passing commands from the OS to the SMI handler. Writes to this register will cause an SMI and set the OS_TCO_SMI bit in the TCO_STS register.

Access Method

Type: IO Register
(Size: 8 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**TCO_DAT_IN (TDI):** Data Register for passing commands from the OS to the SMI handler. Writes to this register will cause an SMI and set the OS_TCO_SMI bit in the TCO_STS register.

7.41 TCO_DAT_OUT Register (TDO)—Offset 3h

Data Register for passing commands from the SMI handler to the OS. Writes to this register will set the TCO_INT_STS bit in the TCO_STS register. It also causes an interrupt, as selected by the TCO_IRQ_SEL bits.

Access Method

Type: IO Register
(Size: 8 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**TCO_DAT_OUT (TDO):** Data Register for passing commands from the SMI handler to the OS. Writes to this register will set the TCO_INT_STS bit in the TCO_STS register. It also causes an interrupt, as selected by the TCO_IRQ_SEL bits.

7.42 TCO1_STS Register (TSTS1)—Offset 4h

Contains the TCO status bits such as NMI2SMI, OS_TCO_SMI, TCO_INT_STS, CPUSCI_STS, CPUSMI_STS etc. Unless otherwise indicated, these bits are sticky and are cleared by writing a 1 to the corresponding bit position.

Access Method

Type: IO Register
(Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RO	**Reserved (RSVD)**
13	0h RO	**TCO Slave Select (TCO_SLVSEL):** This register bit indicates the value of TCO Slave Select Soft Strap.
12	0h RW/1C	**CPUSERR_STS (CPUSERR_STS):** This bit is set to 1 if the CPU complex sends a DMI special cycle message indicating that it wants to cause an SERR#. The software must read the MCH to find out why it wanted the SERR#. Software must write a 1 back to this bit to clear it.
11	0h RO	**Reserved (RSVD_1)**
10	0h RW/1C	**CPUSMI_STS (CPUSMI_STS):** This bit is set to 1 if the CPU complex sends a DMI special cycle message indicating that it wants to cause an SMI. The software must read the CPU to find out why it wanted the SMI. Software must write a 1 back to this bit to clear it.
9	0h RW/1C	**CPUSCI_STS (CPUSCI_STS):** This bit is set to 1 if the CPU complex sends a DMI special cycle message indicating that it wants to cause an SCI. The software must read the CPU to find out why it wanted the SCI. Software must write a 1 back to this bit to clear it.

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW/1C	**BIOSWR_STS (BIOSWR_STS):** SoC sets this bit to 1 and generates and SMI# to indicate an illegal attempt to write to the BIOS. (The BIOS write could be either to FirmwareHUB (FWH) or SPI Flash). This occurs when either: a) The WP bit (10.1.7.4 bit 0) is changed from 0 to 1 and the LE bit (10.1.7.4 bit 1) is also set, or b) any write is attempted to the BIOS and the WP bit is also set. Note: On write cycles attempted to the 4MB lower alias to the BIOS space, the BIOSWR_STS bit will not be set.
7	0h RW/1C	**NEWCENTURY_STS (NEWCENTURY_STS):** This bit will be set when the year rolls over from 1999 to 2000. If the bit is already 1, it will remain 1. This bit can be cleared either by software writing a 1 back to the bit position, or by RTEST# going active. When this bit is set, an SMI# will be generated. However, this will not be a wake event (i.e. if the system is in a sleeping state when the NEWCENTURY_STS bit is set, the system will not wake up). Note: This bit 7 is not valid when the RTC battery is first put in (or if the RTC battery does not provide sufficient power when the system is unplugged). Software can determine that the RTC well was not maintained by checking the RTC_PWR_STS bit (GEN_PMCON_3 register in the Power Management Controller, D31:F2:A4, bit 2) or by other means (such as doing a checksum on the RTC RAM array). If the RTC well is determined to not have been maintained, the BIOS should set the time to a legal value and then clear the NEWCENTURY_STS bit. Note: This bit may take up to 3 RTCCLKs for the bit to be cleared when a 1 is written to the bit to clear it. After writing a 1 to the NEWCENTURY_STS bit, software should also not exit the SMI handler until after the bit has been cleared. This is to make sure the SMI is not re-entered. BIOS Assumption: When booting, the BIOS checks the NEWCENTURY_STS bit. If set, the BIOS should increment the value in the RTC RAM register associated with the century. The BIOS should then clear the NEWCENTURY_STS bit. This scenario would occur if the system was asleep when the century rolls over. If the system is in an S0 state (not sleeping) and the SMI# occurs with the NEWCENTURY_STS bit sets, the SMI handler should increment the value in the RTC RAM register and clear the NEWCENTURY_STS bit.
6:4	0h RO	**Reserved (RSVD_2)**
3	0h RW/1C	**TIMEOUT (TIMEOUT):** Bit set to 1 by SoC to indicate that the SMI was caused by TCO timer reaching 0. Note: The SMI handler should clear this bit to prevent an immediate re-entry to the SMI handler.
2	0h RW/1C	**TCO_INT_STS (TCO_INT_STS):** Bit set to 1 when SMI handler caused the interrupt by writing to the TCO_DAT_OUT register.
1	0h RW/1C	**OS_TCO_SMI (OS_TCO_SMI):** Bit set to 1 when OS code caused an SMI# by writing to the TCO_DAT_IN register.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO/V	**NMI2SMI_STS (NMI2SMI_STS):** The SoC sets read-only this bit when an SMI# occurs because an event occurred that would otherwise have caused an NMI. The NMI2SMI_STS bit should not be sticky bit. It should be a simple OR gate to indicate that one of the NMI sources has caused the SMI#. Each of the NMI sources already has its own sticky bit feeding the OR gate. Writes to this bit have no effect.

7.43 TCO2_STS Register (TSTS2)—Offset 6h

Contains the INTRD_DET, SECOND_TO_STS and SMLINK_SLAVE_SMI_STS bits.

Access Method

Type: IO Register
(Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:5	0h RO	**Reserved (RSVD)**
4	0h RW/1C	**SMLINK_SLAVE_SMI_STS (SMLINK_SLAVE_SMI_STS):** SoC will set this bit to 1 when it receives the SMI message (encoding 08h in the command type) on the SMLinks Slave Interface. Software clears the bit by writing a 1 to this bit position. This bit is in the resume well. It is reset by RSMRST#, but not by the PCI Reset associated with exit from S3-S5 states. This allows the software (presumably BIOS) to get the interrupt, see this new bit set, and decidedly go into the pre-determined (by local policy) sleep state. The advantage here is that race conditions are eliminated if the bit is only meant for power-down instead of potentially being meant for power-up or power-down depending on the current state (like the real power button).
3:2	0h RO	**Reserved (RSVD_1)**
1	0h RW/1C	**SECOND_TO_STS (SECOND_TO_STS):** SoC sets this bit to 1 to indicate that the TIMEOUT bit had been (or is currently) set and a second timeout occurred before the TCO_RLD register was written. If this bit is set and the NO_REBOOT config bit is 0, then the SoC will reboot the system after the second timeout. The reboot is done by asserting PLTRST#. This bit is only cleared by writing a 1 to this bit or by a RSMRST#.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW/1C	**INTRD_DET (INTRD_DET):** Intruder Detect. Bit set to 1 by the SoC to indicate that an intrusion was detected. This is latched. This bit is cleared by writing a 1 to this bit or by RTEST#. This bit is backed in the RTC Well. Note: This bit has a recovery time. After writing a 1 to this bit position (to clear it), the bit may be read back as a 1 for up 65 microseconds before it is read as a 0. Software must be aware of this recovery time when reading this bit after clearing it. Note: If the INTRUDER# signal is active when the software attempts to clear the INTRD_DET bit, the bit will remain as a 1, and the SMI# will be generated again immediately. The SMI handler can clear the INTRD_SEL bits to avoid further SMIs. However, if the INTRUDER# signals goes inactive and then active again, there will not be further SMIs (because the INTRD_SEL bits would select that no SMI# be generated). If the INTRUDER# signal goes inactive some point after the INTRD_DET bit is written as a 1, then the INTRD_DET signal will go to a 0 when INTRUDER# input signal goes inactive. Note that this is slightly different than a classic sticky bit, since most sticky bits would remain active indefinitely when the signal goes active and would immediately go inactive when a 1 is written to the bit.

7.44 TCO1_CNT Register (TCTL1)—Offset 8h

Contains the NMI_NOW, NMI2SMI_EN, TCO_TMR_HALT bits.

Access Method

Type: IO Register (Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	**Reserved (RSVD)**
12	0h RW	**TCO_LOCK (TCO_LOCK):** When set to 1, this bit prevents writes from changing the TCO_EN bit (in offset 30h of Power Management I/O space). Once this bit is set to 1, it can not be cleared by software writing a 0 to this bit location. A core-well reset is required to change this bit from 1 to 0. This bit defaults to 0.
11	0h RW	**TCO_TMR_HALT (TCO_TMR_HALT):** 1 = The TCO timer will halt. It will not count, and thus cannot reach a value that would cause an SMI# or to cause the SECOND_TO_STS bit to be set. This will also prevent rebooting. 0 = The TCO timer is enabled to count. This is the default.
10	0h RW	**Reserved**

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**NMI2SMI_EN (NMI2SMI_EN):** Setting this bit 1 forces all NMIs to instead cause an SMI#, and will be reported in the TCO1_STS register. NMI2SMI_EN bit is set AND the NMI_EN# bit is set to 0, the NMI# will be routed to cause an SMI#. No NMI will be caused. However, if the GBL_SMI_EN bit is not set, then no SMI# will be generated, either. If NMI2SMI_EN is set but the NMI_EN# bit is set to 1, then no NMI or SMI# will be generated. The following table shows the possible combinations: NMI_EN#, GBL_SMI_EN 00 No SMI# based on NMI events (since no SMI# at all because SMI_EN = 0) 01 SMI# will be caused based on NMI events 10 No SMI# at all because SMI_EN is 0 11 No SMI# based on NMI events because NMI_EN#=1
8	0h RW	**NMI_NOW (NMI_NOW):** Writing a 1 to this bit causes an NMI. This allows the BIOS or SMI handler to force entry to the NMI handler. The NMI handler is expected to clear this bit. Another NMI will not be generated until the bit is cleared by writing a 1 back to the same bit position.
7:0	0h RO	**Reserved (RSVD_1)**

7.45 TCO2_CNT Register (TCTL2)—Offset Ah

Contains the INTRD_SEL andf SMBALERT_DISABLE bits.

Access Method

Type: IO Register
(Size: 16 bits)

Device:
Function:

Default: 8h

Bit Range	Default & Access	Field Name (ID): Description
15:6	0h RO	**Reserved (RSVD)**
5:4	0h RW	**OS_POLICY (OS_POLICY):** OS-based software writes to these bits to select the policy that the BIOS will use after the platform resets due the WDT. The following convention is recommended for the BIOS and OS: 00 Boot normally 01 Shut down 10 Dont load OS. Hold in pre-boot state and use LAN to determine next step 11 Reserved Implementation note: These are just scratch pad bits. They should not be reset when the TCO logic resets the platform due to Watchdog Timer.
3	1h RW	**SMB_ALERT_DISABLE (SMB_ALERT_DISABLE):** Disables GP/SMBALERT# as an alert source for the heartbeats and the SMBus slave. At reset (RSMRST# pin assertion only), this bit is set and the muxed GP/SMBALERT# are disabled.

Bit Range	Default & Access	Field Name (ID): Description
2:1	0h RW	**INTRD_SEL (INTRD_SEL):** Selects the action to take if the INTRUDER# signal goes active. 11 Reserved 01 Interrupt (as selected by TCO_INT_SEL). 10 SMI# 00 INTRUDER# doesn't cause SMI# or interrupt
0	0h RO	**Reserved (RSVD_1)**

7.46 TCO_MESSAGE1 and TCO_MESSAGE2 (TMSG)—Offset Ch

BIOS can write into these registers to indicate its boot progress. The external microcontroller can read these registers to monitor the boot progress TCOBASE+0Ch (MSG1) TCOBASE+0Dh (MSG2) BIOS can write into these registers to indicate its boot progress. The external microcontroller can read these registers to monitor the boot progress

Access Method

Type: IO Register
(Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RW	**TCO_MESSAGE2 (MSG2):** BIOS can write into these registers to indicate its boot progress. The external microcontroller can read these registers to monitor the boot progress.
7:0	0h RW	**TCO_MESSAGE1 (MSG1):** BIOS can write into these registers to indicate its boot progress. The external microcontroller can read these registers to monitor the boot progress.

7.47 TCO_WDSTATUS Register (TWDS)—Offset Eh

The BIOS or system management software can write into this register to indicate more details on the boot progress. The register will rest to 00h based on a RSMRST# (but not PCI Reset). The external microcontroller can read this register to monitor boot progress.

Access Method

Type: IO Register
(Size: 8 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**TCO_WDSTATUS Register (TWDS):** The BIOS or system management software can write into this register to indicate more details on the boot progress. The register will rest to 00h based on a RSMRST# (but not PCI Reset). The external microcontroller can read this register to monitor boot progress.

7.48 Reserved_1 (RSV_1)—Offset Fh

Access Method

Type: IO Register
(Size: 8 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Reserved (RSVD)**

7.49 LEGACY_ELIM Register (LE)—Offset 10h

Contains the IRQ1_CAUSE and IRQ2_CAUSE bits.

Access Method

Type: IO Register
(Size: 8 bits)

Device:
Function:

Default: 3h

Bit Range	Default & Access	Field Name (ID): Description
7:2	0h RO	**Reserved (RSVD)**
1	1h RW	**IRQ12_CAUSE (IRQ12_CAUSE):** When software sets the bit to 1, IRQ12 will be high (asserted). When software sets the bit to 0, IRQ12 will be low (not asserted). Default for this bit is 1.
0	1h RW	**IRQ1_CAUSE (IRQ1_CAUSE):** When software sets the bit to 1, IRQ1 will be high (asserted). When software sets the bit to 0, IRQ1 will be low (not asserted). Default for this bit is 1.

7.50 Reserved_2 (RSV_2)—Offset 11h

Access Method

Type: IO Register
(Size: 8 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Reserved (RSVD)**

7.51 TCO_TMR Register (TTMR)—Offset 12h

Value that is loaded into the timer each time the TCO_RLD register is written. The values in this register have no effects on the TCO Timer hardware.

Access Method

Type: IO Register
(Size: 16 bits)

Device:
Function:

Default: 4h

Bit Range	Default & Access	Field Name (ID): Description
15:10	0h RO	**Reserved (RSVD)**
9:0	4h RW	**TCOTMR (TCOTMR):** Value that is loaded into the timer each time the TCO_RLD register is written. Values of 0000h or 0001h will be ignored and should not be attempted. The timer is clocked at approximately 0.6 seconds, and thus allows timeouts ranging from 1.2 second to 613.8 seconds. Note: The timer has an error of +/- 1 tick (0.6s).

7.52 Reserved_3 (RSV_3)—Offset 14h

Access Method

Type: IO Register
(Size: 64 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:0	0h RO	**Reserved (RSVD)**

7.53 Reserved_4 (RSV_4)—Offset 1Ch

Access Method

Type: IO Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

§ §

8 LPC Registers

This chapter documents the registers in Bus: 0, Device 31, Function 0.

8.1 ID — Offset 0h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RO/V	**Device Identification 6 (DID_6):** Device Identification 6 is hardware reset to 9'b000000000. It can be overridden by the SetID IOSF-SB Message.
22:21	0h RO/V	**Device Identification 5 (DID_5):** Device Identification 5 is hardwired to 2'b11.
20	0h RO/V	**Device Identification 4 (DID_4):** Device Identification 4 is set by the dtfus_core_lpcdid[4] fuse.
19	0h RO/V	**Device Identification 3 (DID_3):** Device Identification 3 is set by the dtfus_core_lpcdid[3] fuse.
18	0h RO/V	**Device Identification 2 (DID_2):** Device Identification 2 is set by the dtfus_core_lpcdid[2] fuse.
17	0h RO/V	**Device Identification 1 (DID_1):** Device Identification 1 is set by the dtfus_core_lpcdid[1] fuse.
16	0h RO/V	**Device Identification 0 (DID_0):** Device Identification 0 is set by the dtfus_core_lpcdid[0] fuse.
15:0	0h RO	**Vendor Identification (VID):** Indicates Intel.

8.2 Device Command (CMD) — Offset 4h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:31, F:0] + 4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:11	0h RO	**Reserved (RSVD):** Reserved
10	0h RO	**Interrupt Disable (ID):** The LPC bridge has no interrupts to disable.
9	0h RO	**Fast Back to Back Enable (FBE):** Reserved as 0 per PCI-Express spec.
8	0h RW	**SERR# Enable (SEE):** The LPC bridge generates SERR# if this bit is set.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**Wait Cycle Control (WCC):** Reserved as 0 per PCI-Express spec.
6	0h RW	**Parity Error Response Enable (PERE):** When this bit is set to 1, it enables the LPC bridge to response to parity errors detected on backbone interface.
5	0h RO	**VGA Palette Snoop (VGA_PSE):** Reserved as 0 per PCI-Express spec.
4	0h RO	**Memory Write and Invalidate Enable (MWIE):** Reserved as 0 per PCI-Express spec.
3	0h RO	**Special Cycle Enable (SCE):** Reserved as 0 per PCI-Express spec.
2	0h RO	**Bus Master Enable (BME):** Bus Masters cannot be disabled.
1	0h RO	**Memory Space Enable (MSE):** Memory space cannot be disabled on LPC.
0	0h RO	**I/O Space Enable (IOSE):** I/O space cannot be disabled on LPC.

8.3 Status (STS) — Offset 6h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:31, F:0] + 6h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**Detected Parity Error (DPE):** Set when the bridge detects a parity error on the internal backbone. This bit gets set even if CMD.PERE is not set.
14	0h RW/1C	**Signaled System Error (SSE):** Set when the LPC bridge signals a system error to the internal SERR# logic.
13	0h RO	**Received Master Abort (RMA):** Set when the bridge receives a completion with unsupported request status from the backbone. LPC is a target only controller.
12	0h RO	**Received Target Abort (RTA):** Set when the bridge receives a completion with completer abort status from the backbone. LPC is a target only controller.
11	0h RW/1C	**Signalled Target Abort (STA):** Set when the bridge generates a completion packet with target abort status on the backbone.
10:9	0h RO	**DEVSEL# Timing Status (DTS):** Indicates medium timing, although this has no meaning on the backbone.
8	0h RW/1C	**Data Parity Error Detected (DPD):** Set when the bridge receives a completion packet from the backbone from a previous request, and detects a parity error, and CMD.PERE is set.
7	0h RO	**Fast Back to Back Capable (FBC):** Reserved - bit has no meaning on internal backbone.
6	0h RO	**Reserved (RSVD):** Reserved
5	0h RO	**66 MHz Capable (C66):** Reserved - bit has no meaning on internal backbone.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RO	**Capabilities List (CLIST):** There is a capabilities list in the LPC bridge.
3	0h RO	**Interrupt Status (IS):** The LPC bridge does not generate interrupts.
2:0	0h RO	**Reserved (RSVD_1):** Reserved

8.4 Revision ID (RID) — Offset 8h

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:31, F:0] + 8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO/V	**Revision ID (RID):** Indicates the part revision. This will reset to 0 but is expected to be overridden by the SetID IOSF-SB message.

8.5 Class Code (CC) — Offset 9h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 9h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23:16	0h RO	**Base Class Code (BCC):** Indicates the device is a bridge device.
15:8	0h RO	**Sub-Class Code (SCC):** Indicates the device a PCI to ISA bridge.
7:0	0h RO	**Programming Interface (PI):** The LPC bridge has no programming interface.

8.6 Primary Latency Timer (PLT) — Offset Dh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:31, F:0] + Dh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:3	0h RO	**Master Latency Count (MLC):** Reserved per 3GIO spec.
2:0	0h RO	**Reserved (RSVD):** Reserved

8.7 Header Type (HTYPE) — Offset Eh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:31, F:0] + Eh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**Multi-function Device (MFD):** This bit is 1 to indicate a multifunction device.
6:0	0h RO	**Header Type (HTYPE):** Identifies the header layout of the configuration space, which is a generic device.

8.8 Sub System Identifiers (SS) — Offset 2Ch

This register is initialized to logic 0 by the assertion of PLTRST#. This register can be written only once after PLTRST# de-assertion.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**Subsystem ID (SSID):** This is written by BIOS. No hardware action taken on this value.
15:0	0h RW/O	**Subsystem Vendor ID (SSVID):** This is written by BIOS. No hardware action taken on this value.

8.9 Capability List Pointer (CAPP) — Offset 34h

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:31, F:0] + 34h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Capability Pointer (CP):** Indicates the offset of the first Capability Item.

8.10 Serial IRQ Control (SCNT) — Offset 64h

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:31, F:0] + 64h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**EN:** When set, serial IRQs will be recognized.
6	0h RW	**Mode (MD):** When set, the serial IRQ machine will be in continuous mode. When cleared, the serial IRQ machine will be in quiet mode. When setting the EN bit, this bit must also be written as a one to guarantee that the first action of the serial IRQ machine will be a start frame.
5:2	0h RO	**Frame Size (FS):** Fixed field that indicates the size of the SERIRQ frame as 21 frames.
1:0	0h RW	**Start Frame Pulse Width (SFPW):** This is the number of 33 MHz clocks that the SERIRQ pin will be driven low by the Serial IRQ controller to signal a start frame. In continuous mode, the controller will drive the start frame for the number of clocks specified. In quiet mode, the controller will drive the start frame for the number of clocks specified minus one, as the first clock was driven by the peripheral. Bits Clocks • 00: 4 • 01: 6 • 10: 8 • 11: Reserved

8.11 B-Unit Copy of the I/O Decode Ranges Register for LPC (B_CR_IOD_IOE_LPC) — Offset 80h

Affects B-Unit SAD for LPC regions.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 80h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (RSVD):** Reserved
25	0h RW/V	**High Gameport Enable (HGE):** Enables decoding of the I/O locations 208h to 20Fh to LPC.
24	0h RW/V	**Low Gameport Enable (LGE):** Enables decoding of the I/O locations 200h to 207h to LPC.

Bit Range	Default & Access	Field Name (ID): Description
23:20	0h RO	**Reserved (RSVD):** Reserved
19	0h RW/V	**Floppy Drive Enable (FDE):** Enables decoding of the FDD range to LPC. Range is selected by LIOD.FDE.
18	0h RW/V	**Parallel Port Enable (PPE):** Enables decoding of the LPT range to LPC. Range is selected by LIOD.LPT.
17	0h RW/V	**Com Port B Enable (CBE):** Enables decoding of the ComB range to LPC. Range is selected LIOD.CB.
16	0h RW/V	**Com Port A Enable (CAE):** Enables decoding of the ComA range to LPC. Range is selected LIOD.CA.
15:13	0h RO	**Reserved (RSVD):** Reserved
12	0h RW/V	**FDD:** The following table describes which range to decode for the FDD Port. • 0 => 3F0h-3F5h, 3F7h (Primary) • 1 => 370h-375h, 377h (Secondary)
11:10	0h RO	**Reserved (RSVD):** Reserved
9:8	0h RW/V	**LPT:** The following table describes which range to decode for the LPT Port. • 00: 378h-37Fh and 778h-77Fh • 01: 278h-27Fh (port 279h is read only) and 678h-67Fh • 10: 3BCh-3BEh and 7BCh-7BEh • 11: Reserved
7	0h RO	**Reserved (RSVD):** Reserved
6:4	0h RW/V	**ComB Range (CB):** The following table describes which range to decode for the ComB Port. • 000: 3F8h-3FFh (COM 1) • 001: 2F8h-2FFh (COM 2) • 010: 220h-227h • 011: 228h-22Fh • 100: 238h-23Fh • 101: 2E8h-2EFh (COM 4) • 110: 338h-33Fh • 111: 3E8h-3EFh (COM 3)
3	0h RO	**Reserved (RSVD):** Reserved
2:0	0h RW/V	**ComA Range (CA):** The following table describes which range to decode for the ComA Port. • 000: 3F8h-3FFh (COM 1) • 001: 2F8h-2FFh (COM 2) • 010: 220h-227h • 011: 228h-22Fh • 100: 238h-23Fh • 101: 2E8h-2EFh (COM 4) • 110: 338h-33Fh • 111: 3E8h-3EFh (COM 3)

8.12 I/O Enables (IOE) — Offset 82h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:31, F:0] + 82h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RO	**Reserved (RSVD):** Reserved
13	0h RW	**Microcontroller Enable #2 (ME2):** Enables decoding of I/O locations 4Eh and 4Fh to LPC.
12	0h RW	**SuperI/O Enable (SE):** Enables decoding of I/O locations 2Eh and 2Fh to LPC.
11	0h RW	**Microcontroller Enable #1 (ME1):** Enables decoding of I/O locations 62h and 66h to LPC.
10	0h RW	**KE:** Enables decoding of the keyboard I/O locations 60h and 64h to LPC.
9	0h RW	**High Gameport Enable (HGE):** Enables decoding of the I/O locations 208h to 20Fh to LPC.
8	0h RW	**Low Gameport Enable (LGE):** Enables decoding of the I/O locations 200h to 207h to LPC.
7:4	0h RO	**Reserved (RSVD_1):** Reserved
3	0h RW	**Floppy Drive Enable (FDE):** Enables decoding of the FDD range to LPC. Range is selected by LIOD.FDE
2	0h RW	**Parallel Port Enable (PPE):** Enables decoding of the LPT range to LPC. Range is selected by LIOD.LPT.
1	0h RW	**Com Port B Enable (CBE):** Enables decoding of the COMB range to LPC. Range is selected LIOD.CB.
0	0h RW	**Com Port A Enable (CAE):** Enables decoding of the COMA range to LPC. Range is selected LIOD.CA.

8.13 B_LGIR1_LPC (B_CR_LGIR1_LPC) — Offset 84h

B-Unit copy of the LPC Generic I/O Range #1 register for LPC. Affects B-Unit SAD for LPC regions.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 84h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
23:18	0h RW/V	**Address [7:2] Mask (ADDR_MASK_7_2):** A '1' in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match. The corresponding bit in the Address field, below, is ignored. The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size.
17:16	0h RO	**Reserved (RSVD):** Reserved
15:2	0h RW/V	**Address [15:2] (ADDR_15_2):** DWord-aligned address. Note that PCH does not provide decode down to the word or byte level.
1	0h RO	**Reserved (RSVD):** Reserved
0	0h RW/V	**LPC Decode Enable (EN):** When this bit is set to '1', then the range specified in this register is enabled for decoding to LPC.

8.14 B_LGIR2_LPC (B_CR_LGIR2_LPC) — Offset 88h

B-Unit copy of the LPC Generic I/O Range #2 register for LPC. Affects B-Unit SAD for LPC regions.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 88h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23:18	0h RW/V	**Address [7:2] Mask (ADDR_MASK_7_2):** A '1' in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match. The corresponding bit in the Address field, below, is ignored. The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size.
17:16	0h RO	**Reserved (RSVD):** Reserved
15:2	0h RW/V	**Address [15:2] (ADDR_15_2):** DWord-aligned address. Note that PCH does not provide decode down to the word or byte level.
1	0h RO	**Reserved (RSVD):** Reserved
0	0h RW/V	**LPC Decode Enable (EN):** When this bit is set to '1', then the range specified in this register is enabled for decoding to LPC.

8.15 B_LGIR3_LPC (B_CR_LGIR3_LPC) — Offset 8Ch

B-Unit copy of the LPC Generic I/O Range #3 register for LPC. Affects B-Unit SAD for LPC regions.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 8Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23:18	0h RW/V	**Address [7:2] Mask (ADDR_MASK_7_2):** A '1' in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match. The corresponding bit in the Address field, below, is ignored. The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size.
17:16	0h RO	**Reserved (RSVD):** Reserved
15:2	0h RW/V	**Address [15:2] (ADDR_15_2):** DWord-aligned address. Note that PCH does not provide decode down to the word or byte level.
1	0h RO	**Reserved (RSVD):** Reserved
0	0h RW/V	**LPC Decode Enable (EN):** When this bit is set to '1', then the range specified in this register is enabled for decoding to LPC.

8.16 B_LGIR4_LPC (B_CR_LGIR4_LPC) — Offset 90h

B-Unit copy of the LPC Generic I/O Range #4 register for LPC. Affects B-Unit SAD for LPC regions.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 90h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23:18	0h RW/V	**Address [7:2] Mask (ADDR_MASK_7_2):** A '1' in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match. The corresponding bit in the Address field, below, is ignored. The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size.
17:16	0h RO	**Reserved (RSVD):** Reserved
15:2	0h RW/V	**Address [15:2] (ADDR_15_2):** DWord-aligned address. Note that PCH does not provide decode down to the word or byte level.
1	0h RO	**Reserved (RSVD):** Reserved
0	0h RW/V	**LPC Decode Enable (EN):** When this bit is set to '1', then the range specified in this register is enabled for decoding to LPC.

8.17 USB Legacy Keyboard/Mouse Control (ULKMC) — Offset 94h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 94h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15	0h RW/1C	**SMI Caused by End of Pass-through (SMIBYENDPS):** Indicates if the event occurred. Note that even if the corresponding enable bit is not set in the Bit 7, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Writing a 1 to this bit will clear the latch.
14	0h RO	**Reserved (RSVD_1):** Reserved
13	0h RW	**Reserved (RSVD_2):** Reserved
12	0h RO	**Reserved (RSVD_3):** Reserved
11	0h RW/1C	**SMI Caused by Port 64 Write (TRAPBY64W):** Indicates if the event occurred. Note that even if the corresponding enable bit is not set in the Bit 3, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Writing a 1 to this bit will clear the latch. Note that the A20Gate Pass-Through Logic allows specific port 64h Writes to complete without setting this bit.
10	0h RW/1C	**SMI Caused by Port 64 Read (TRAPBY64R):** Indicates if the event occurred. Note that even if the corresponding enable bit is not set in the Bit 2, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Writing a 1 to this bit will clear the latch.
9	0h RW/1C	**SMI Caused by Port 60 Write (TRAPBY60W):** Indicates if the event occurred. Note that even if the corresponding enable bit is not set in the Bit 1, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Writing a 1 to this bit will clear the latch. Note that the A20Gate Pass-Through Logic allows specific port 60h Writes to complete without setting this bit.
8	0h RW/1C	**SMI Caused by Port 60 Read (TRAPBY60R):** Indicates if the event occurred. Note that even if the corresponding enable bit is not set in the Bit 0, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Writing a 1 to this bit will clear the latch.
7	0h RW	**SMI at End of Pass-through Enable (SMIATENDPS):** May need to cause SMI at the end of a pass-through. Can occur if an SMI is generated in the middle of a pass through, and needs to be serviced later.
6	0h RO	**Pass Through State (PSTATE):** This read-only bit indicates that the state machine is in the middle of an A20GATE pass-through sequence. If software needs to reset this bit, it should set Bit 5 0.
5	0h RW	**A20Gate Pass-Through Enable (A20PASSEN):** When enabled, allows A20GATE sequence Pass-Through function. When enabled, a specific cycle sequence involving writes to port 60h and port 64h does not result in the setting of the SMI status bits.SMI# will not be generated, even if the various enable bits are set.
4	0h RW	**Reserved (RSVD_4):** Reserved
3	0h RW	**SMI on Port 64 Writes Enable (S64WEN):** When set, a 1 in bit 11 will cause an SMI event.
2	0h RW	**SMI on Port 64 Reads Enable (S64REN):** When set, a 1 in bit 10 will cause an SMI event.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**SMI on Port 60 Writes Enable (S60WEN):** When set, a 1 in bit 9 will cause an SMI event.
0	0h RW	**SMI on Port 60 Reads Enable (S60REN):** When set, a 1 in bit 8 will cause an SMI event.

8.18 B-Unit Copy of the LPC Generic Memory Range Register for LPC (B_CR_LGMR_LPC) — Offset 98h

Affects B-Unit SAD for LPC regions.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 98h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/V	**Memory Address [31:16] (ADDR_31_16):** This field specifies a 64KB memory block anywhere in the 4GB memory space that will be decoded to LPC as standard LPC Memory Cycle if enabled.
15:1	0h RO	**Reserved (RSVD):** Reserved
0	0h RW/V	**LPC Memory Range Decode Enable (EN):** When this bit is set to '1', then the range specified in this register is enabled for decoding to LPC.

8.19 Reserved (RSVD) — Offset 9Ch

Reserved.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 9Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD):** Reserved

8.20 B-Unit Copy of the eSPI CS1# Generic I/O Range #1 (B_CR_PCCS1GIR1_ESPI) — Offset A4h

Affects B-Unit SAD for LPC regions.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + A4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23:18	0h RW/V	**Address [7:2] Mask (ADDR_MASK_7_2):** A '1' in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match. The corresponding bit in the Address field, below, is ignored. The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size.
17:16	0h RO	**Reserved (RSVD):** Reserved
15:2	0h RW/V	**Address[15:2] (ADDR_15_2):** DWord-aligned address. Note: PCH does not provide decode down to the word or byte level.
1	0h RO	**Reserved (RSVD):** Reserved
0	0h RW/V	**LPC Decode Enable (EN):** When this bit is set to '1', then the range specified in this register is enabled for decoding to LPC (eSPI).

8.21 B-Unit Copy of the eSPI CS1# Generic Memory Range #1 Register (B_CR_PCCS1GMR1_ESPI) — Offset A8h

Affects B-Unit SAD for LPC regions.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + A8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/V	**Memory Address [31:16] (ADDR_31_16):** This field specifies a 64KB memory block anywhere in the 4GB memory space that will be decoded to LPC (eSPI) as standard LPC Memory Cycle, if enabled.
15:1	0h RO	**Reserved (RSVD):** Reserved
0	0h RW/V	**LPC Memory Range Decode Enable (EN):** When this bit is set to '1', then the range specified in this register is enabled for decoding to LPC (eSPI).

8.22 FWH ID Select #1 (FS1) — Offset D0h

This register contains the IDSEL fields the LPC Bridge uses for memory cycles going to the FWH.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + D0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**F8-FF IDSEL (IF8):** IDSEL to use in FWH cycle for range enabled by BDE.EF8.
27:24	0h RW	**F0-F7 IDSEL (IF0):** IDSEL to use in FWH cycle for range enabled by BDE.EF0.
23:20	0h RW	**E8-EF IDSEL (IE8):** IDSEL to use in FWH cycle for range enabled by BDE.EE8.
19:16	0h RW	**E0-E7 IDSEL (IE0):** IDSEL to use in FWH cycle for range enabled by BDE.EE0.
15:12	0h RW	**D8-DF IDSEL (ID8):** IDSEL to use in FWH cycle for range enabled by BDE.ED8.
11:8	0h RW	**D0-D7 IDSEL (ID0):** IDSEL to use in FWH cycle for range enabled by BDE.ED0.
7:4	0h RW	**C8-CF IDSEL (IC8):** IDSEL to use in FWH cycle for range enabled by BDE.EC8.
3:0	0h RW	**C0-C7 IDSEL (IC0):** IDSEL to use in FWH cycle for range enabled by BDE.EC0.

8.23 FWH ID Select #2 (FS2) — Offset D4h

This register contains the additional IDSEL fields the LPC Bridge uses for memory cycles going to the FWH.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:31, F:0] + D4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:12	0h RW	**70-7F IDSEL (I70):** IDSEL to use in FWH cycle for range enabled by BDE.E70.
11:8	0h RW	**60-6F IDSEL (I60):** IDSEL to use in FWH cycle for range enabled by BDE.E60.
7:4	0h RW	**50-5F IDSEL (I50):** IDSEL to use in FWH cycle for range enabled by BDE.E50.
3:0	0h RW	**40-4F IDSEL (I40):** IDSEL to use in FWH cycle for range enabled by BDE.E40.

8.24 B-Unit Copy of the BIOS Decode Enable Register for LPC (B_CR_BDE_LPC) — Offset D8h

Affects B-Unit SAD for BIOS regions only when BIOS is resident in LPC.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + D8h	FFCF h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RESERVED_0):** Reserved
15	1h RO	**F8-FF Enable (EF8):** Enables decoding of 512K of the following BIOS ranges: FFF80000h-FFFFFFFFh and FFB80000h-FFBFFFFFh.
14	1h RW/V	**F0-F8 Enable (EF0):** Enables decoding of 512K of the following BIOS ranges: FFF00000h-FFF7FFFFh and FFB00000h-FFB7FFFFh.
13	1h RW/V	**E8-EF Enable (EE8):** Enables decoding of 512K of the following BIOS ranges: FFE80000h-FFEFFFFFh and FFA80000h-FFAFFFFFh.
12	1h RW/V	**E0-E8 Enable (EE0):** Enables decoding of 512K of the following BIOS ranges: FFE00000h-FFE7FFFFh and FFA00000h-FFA7FFFFh.
11	1h RW/V	**D8-DF Enable (ED8):** Enables decoding of 512K of the following BIOS ranges: FFD80000h-FFDFFFFFh and FF980000h-FF9FFFFFh.
10	1h RW/V	**D0-D7 Enable (ED0):** Enables decoding of 512K of the following BIOS ranges: FFD00000h-FFD7FFFFh and FF900000h-FF97FFFFh.
9	1h RW/V	**C8-CF Enable (EC8):** Enables decoding of 512K of the following BIOS ranges: FFC80000h-FFCFFFFFh and FF880000h-FF8FFFFFh.
8	1h RW/V	**C0-C7 Enable (EC0):** Enables decoding of 512K of the following BIOS ranges: FFC00000h-FFC7FFFFh and FF800000h-FF87FFFFh.
7	1h RW/V	**Legacy F Segment Enable (LFE):** This enables the decoding of the legacy 64KB range at F0000h-FFFFFh.
6	1h RW/V	**Legacy E Segment Enable (LEE):** This enables the decoding of the legacy 64KB range at E0000h-EFFFFh.
5:4	0h RO	**Reserved (RESERVED_1):** Reserved
3	1h RW/V	**70-7F Enable (E70):** Enables decoding of 1MB of the following BIOS ranges: FF700000h-FF7FFFFFh and FF300000h-FF3FFFFFh.
2	1h RW/V	**60-6F Enable (E60):** Enables decoding of 1MB of the following BIOS ranges: FF600000h-FF6FFFFFh and FF200000h-FF2FFFFFh.
1	1h RW/V	**50-5F Enable (E50):** Enables decoding of 1MB of the following BIOS ranges: FF500000h-FF5FFFFFh and FF100000h-FF1FFFFFh.
0	1h RW/V	**40-4F Enable (E40):** Enables decoding of 1MB of the following BIOS ranges: FF400000h-FF4FFFFFh and FF000000h-FF0FFFFFh.

8.25 BIOS Control (BC) — Offset DCh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:31, F:0] + DCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW/1L	**BIOS Interface Lock-Down (BILD):** When set, prevents BC.TS and BC.BBS from being changed. This bit can only be written from 0 to 1 once. BIOS Note: This bit is not backed up in the RTC well. This bit should also be set in the BUC register in the RTC device to record the last state of this value following a cold reset.
6	0h RW/L	**Boot BIOS Strap (BBS):** This field determines the destination of accesses to the BIOS memory range. • 0: SPI • 1: LPC When SPI or LPC is selected, the range that is decoded is further qualified by other configuration bits. The value in this field can be overwritten by software as long as the BIOS Interface Lock-Down is not set.
5	0h RW/L	**Enable InSMM.STS (EISS):** When this bit is set, the BIOS region is not writable until SMM sets the InSMM.STS bit. Today BIOS Flash is writable if WPD is a 1. If this bit [5] is set, then WPD must be a 1 and InSMM.STS (0xFED3_0880[0]) must be 1 also. If this bit [5] is clear, then BIOS is writable based only on WPD = 1 and the InSMM.STS is a don't care.
4	0h RO	**Top Swap (TS):** When set, PCH will invert either A16, A17, or A18 for cycles going to the BIOS space (but not the feature space) in the FWH. When cleared, PCH will not invert A16. If booting from LPC (FWH), then the Boot Block size is 64KB and A16 is inverted if Top Swap is enabled. If booting from SPI, then the BOOT_BLOCK_SIZE soft strap determines if A16, A17, or A18 should be inverted if Top Swap is enabled. *If PCH is strapped for Top-Swap (GNT[3]# is low at rising edge of PWROK), then this bit cannot be cleared by software. The strap jumper should be removed and the system rebooted. BIOS Note: [list=1]

8.26 RBR (RBR)—Offset 0h

Receive Buffer Register

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h NA	**Res_rbr (Res_rbr)**
7:0	0h RO	**rbr (rbr)**

8.27 THR (THR)—Offset 0h

Transmit Holding Register

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h NA	**Res_thr (Res_thr)**
7:0	0h WO	**thr (thr)**

8.28 DLL (DLL)—Offset 0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h NA	**Res_dll (Res_dll)**
7:0	0h RW	**dll (dll)**

8.29 IER (IER)—Offset 4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h NA	**Res_31_8 (Res_31_8)**
7	0h RW	**PTIME (PTIME)**
6:4	0h NA	**Res_6_4 (Res_6_4)**
3	0h RW	**EDSSI (EDSSI)**
2	0h RW	**ELSI (ELSI)**
1	0h RW	**ETBEI (ETBEI)**
0	0h RW	**ERBFI (ERBFI)**

8.30 DLH (DLH)—Offset 4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h NA	**Res_31_8 (Res_31_8)**
7:0	0h RW	**dlh (dlh)**

8.31 IIR (IIR)—Offset 8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h NA	**Res_31_8 (Res_31_8)**
7:6	0h RO	**FIFOSE (FIFOSE)**
5:4	0h NA	**Res_5_4 (Res_5_4)**
3:0	1h RO	**IID (IID)**

8.32 FCR (FCR)—Offset 8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h NA	**Res_31_8 (Res_31_8)**
7:6	0h WO	**RCVR (RCVR)**

Bit Range	Default & Access	Field Name (ID): Description
5:4	0h WO	**TET (TET)**
3	0h WO	**DMAM (DMAM)**
2	0h WO	**XFIFOR (XFIFOR)**
1	0h WO	**RFIFOR (RFIFOR)**
0	1h WO	**FIFOE (FIFOE)**

8.33 LCR (LCR)—Offset Ch

Line Control Register

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7	0h RW	**DLAB (DLAB):** Divisor Latch Access Bit. If UART_16550_COMPATIBLE == NO, then writeable only when UART is not busy (USR[0] is zero); otherwise always writable, always readable. This bit is used to enable reading and writing of the Divisor Latch register (DLL and DLH) to set the baud rate of the UART. This bit must be cleared after initial baud rate setup in order to access other registers. Reset Value: 0x0
6	0h RW	**Break (Break):** Break Control Bit.This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE == Enabled and active (MCR[6] set to one) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver and the sir_out_n line is forced low. Reset Value: 0x0
5	0h RO	Reserved.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW	**EPS (EPS):** Even Parity Select. If UART_16550_COMPATIBLE == NO, then writeable only when UART is not busy (USR[0] is zero); otherwise always writable, always readable. This is used to select between even and odd parity, when parity is enabled (PEN set to one). If set to one, an even number of logic 1s is transmitted or checked. If set to zero, an odd number of logic 1s is transmitted or checked. Reset Value: 0x0
3	0h RW	**PEN (PEN):** Parity Enable. If UART_16550_COMPATIBLE == NO, then writeable only when UART is not busy (USR[0] is zero); otherwise always writable, always readable. This bit is used to enable and disable parity generation and detection in transmitted and received serial character respectively. 0 = parity disabled 1 = parity enabled Reset Value: 0x0
2	0h RW	**STOP (STOP):** Number of stop bits. If UART_16550_COMPATIBLE == NO, then writeable only when UART is not busy (USR[0] is zero); otherwise always writable, always readable. This is used to select the number of stop bits per character that the peripheral transmits and receives. If set to zero, one stop bit is transmitted in the serial data. If set to one and the data bits are set to 5 (LCR[1:0] set to zero) one and a half stop bits is transmitted. Otherwise, two stop bits are transmitted. Note that regardless of the number of stop bits selected, the receiver checks only the first stop bit. 0 = 1 stop bit 1 = 1.5 stop bits when DLS (LCR[1:0]) is zero, else 2 stop bit Reset Value: 0x0
1:0	0h RW	**DLS (DLS):** Data Length Select. If UART_16550_COMPATIBLE == NO, then writeable only when UART is not busy (USR[0] is zero); otherwise always writable, always readable. This is used to select the number of data bits per character that the peripheral transmits and receives. The number of bit that may be selected areas follows: 00 = 5 bits 01 = 6 bits 10 = 7 bits 11 = 8 bits Reset Value: 0x0

8.34 MCR (MCR)—Offset 10h

Modem Control Register

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RO	Reserved.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**AFCE (AFCE):** Auto Flow Control Enable. Writeable only when AFCE_MODE == Enabled, always readable. When FIFOs are enabled and the Auto Flow Control Enable (AFCE) bit is set, Auto Flow Control features are enabled as described in Auto Flow Control on page 37. 0 = Auto Flow Control Mode disabled 1 = Auto Flow Control Mode enabled Reset Value: 0x0
4	0h RW	**LoopBack (LoopBack):** LoopBack Bit. This is used to put the UART into a diagnostic mode for test purposes. If operating in UART mode (SIR_MODE != Enabled or not active, MCR[6] set to zero), data on the sout line is held high, while serial data output is looped back to the sin line, internally. In this mode all the interrupts are fully functional. Also, in loopback mode, the modem control inputs (dsr_n, cts_n, ri_n, dcd_n) are disconnected and the modem control outputs (dtr_n, rts_n, out1_n, out2_n) are looped back to the inputs, internally. If operating in infrared mode (SIR_MODE == Enabled AND active, MCR[6] set to one), data on the sir_out_n line is held low, while serial data output is inverted and looped back to the sir_in line. Reset Value: 0x0
3	0h RW	**OUT2 (OUT2):** OUT2. This is used to directly control the user-designated Output2 (out2_n) output. The value written to this location is inverted and driven out on out2_n, that is: 0 = out2_n de-asserted (logic 1) 1 = out2_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to one), the out2_n output is held inactive high while the value of this location is internally looped back to an input. Reset Value: 0x0
2	0h RW	**OUT1 (OUT1):** OUT1. This is used to directly control the user-designated Output1 (out1_n) output. The value written to this location is inverted and driven out on out1_n, that is: 0 = out1_n de-asserted (logic 1) 1 = out1_n asserted (logic 0) Note that in Loopback mode (MCR[4] set to one), the out1_n output is held inactive high while the value of this location is internally looped back to an input. Reset Value: 0x0
1	0h RW	**RTS (RTS):** Request to Send. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the UART is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] set to zero), the rts_n signal is set low by programming MCR[1] (RTS) to a high.In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] set to one) and FIFOs enable (FCR[0] set to one), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). The rts_n signal is de-asserted when MCR[1] is set low. Note that in Loopback mode (MCR[4] set to one), the rts_n output is held inactive high while the value of this location is internally looped back to an input. Reset Value: 0x0

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**DTR (DTR):** Data Terminal Ready. This is used to directly control the Data Terminal Ready (dtr_n) output. The value written to this location is inverted and driven out on dtr_n, that is: 0 = dtr_n de-asserted (logic 1) 1 = dtr_n asserted (logic 0) The Data Terminal Ready output is used to inform the modem or data set that the UART is ready to establish communications. Note that in Loopback mode (MCR[4] set to one), the dtr_n output is held inactive high while the value of this location is internally looped back to an input. Reset Value: 0x0

8.35 LSR (LSR)—Offset 14h

Line Status Register

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 60h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7	0h RW	**RFE (RFE):** Receiver FIFO Error bit. This bit is only relevant when FIFO_MODE != NONE AND FIFOs are enabled (FCR[0] set to one). This is used to indicate if there is at least one parity error, framing error, or break indication in the FIFO. 0 = no error in RX FIFO 1 = error in RX FIFO This bit is cleared when the LSR is read and the character with the error is at the top of the receiver FIFO and there are no subsequent errors in the FIFO. Reset Value: 0x0
6	1h RW	**TEMT (TEMT):** Transmitter Empty bit. If in FIFO mode (FIFO_MODE != NONE) and FIFOs enabled (FCR[0] set to one), this bit is set whenever the Transmitter Shift Register and the FIFO are both empty. If in non-FIFO mode or FIFOs are disabled, this bit is set whenever the Transmitter Holding Register and the Transmitter Shift Register are both empty. Reset Value: 0x1

Bit Range	Default & Access	Field Name (ID): Description
5	1h RW	**THRE (THRE):** Transmit Holding Register Empty bit. If THRE_MODE_USER == Disabled or THRE mode is disabled (IER[7] set to zero) and regardless of FIFO's being implemented/enabled or not, this bit indicates that the THR or TX FIFO is empty. This bit is set whenever data is transferred from the THR or TX FIFO to the transmitter shift register and no new data has been written to the THR or TX FIFO. This also causes a THRE Interrupt to occur, if the THRE Interrupt is enabled. If THRE_MODE_USER == Enabled AND FIFO_MODE != NONE and both modes are active (IER[7] set to one and FCR[0] set to one respectively), the functionality is switched to indicate the transmitter FIFO is full, and no longer controls THRE interrupts, which are then controlled by the FCR[5:4] threshold setting. For more details, see Programmable THRE Interrupt on page 40. Reset Value: 0x1
4	0h RW	**BI (BI):** Break Interrupt bit. This is used to indicate the detection of a break sequence on the serial input data. If in UART mode (SIR_MODE == Disabled), it is set whenever the serial input, sin, is held in a logic '0' state for longer than the sum of start time + data bits + parity + stop bits. If in infrared mode (SIR_MODE == Enabled), it is set whenever the serial input, sir_in, is continuously pulsed to logic '0' for longer than the sum of start time + data bits + parity + stop bits. A break condition on serial input causes one and only one character, consisting of all zeros, to be received by the UART. In the FIFO mode, the character associated with the break condition is carried through the FIFO and is revealed when the character is at the top of the FIFO. Reading the LSR clears the BI bit. In the non-FIFO mode, the BI indication occurs immediately and persists until the LSR is read. Reset Value: 0x0
3	0h RW	**FE (FE):** Framing Error bit. This is used to indicate the occurrence of a framing error in the receiver. A framing error occurs when the receiver does not detect a valid STOP bit in the received data. In the FIFO mode, since the framing error is associated with a character received, it is revealed when the character with the framing error is at the top of the FIFO. When a framing error occurs, the UART tries to resynchronize. It does this by assuming that the error was due to the start bit of the next character and then continues receiving the other bit i.e. data, and/or parity and stop. It should be noted that the Framing Error (FE) bit (LSR[3]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0 = no framing error 1 = framing error Reading the LSR clears the FE bit. Reset Value: 0x0
2	0h RW	**PE (PE):** Parity Error bit. This is used to indicate the occurrence of a parity error in the receiver if the Parity Enable (PEN) bit (LCR[3]) is set. In the FIFO mode, since the parity error is associated with a character received, it is revealed when the character with the parity error arrives at the top of the FIFO. It should be noted that the Parity Error (PE) bit (LSR[2]) is set if a break interrupt has occurred, as indicated by Break Interrupt (BI) bit (LSR[4]). 0 = no parity error 1 = parity error Reading the LSR clears the PE bit. Reset Value: 0x0

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**OE (OE):** Overrun error bit. This is used to indicate the occurrence of an overrun error. This occurs if a new data character was received before the previous data was read. In the non-FIFO mode, the OE bit is set when a new character arrives in the receiver before the previous character was read from the RBR. When this happens, the data in the RBR is overwritten. In the FIFO mode, an overrun error occurs when the FIFO is full and a new character arrives at the receiver. The data in the FIFO is retained and the data in the receive shift register is lost. 0 = no overrun error 1 = overrun error Reading the LSR clears the OE bit. Reset Value: 0x0
0	0h RW	**DR (DR):** Data Ready bit. This is used to indicate that the receiver contains at least one character in the RBR or the receiver FIFO. 0 = no data ready 1 = data ready This bit is cleared when the RBR is read in non-FIFO mode, or when the receiver FIFO is empty, in FIFO mode. Reset Value: 0x0

8.36 MSR (MSR)—Offset 18h

Modem Status Register

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7	0h RO	**DCD (DCD)**
6	0h RO	**RI (RI)**
5	0h RO	**DSR (DSR)**
4	0h RO	**CTS (CTS)**
3	0h RO	**DDCD (DDCD)**
2	0h RO	**TERI (TERI)**
1	0h RO	**DDSR (DDSR)**
0	0h RO	**DCTS (DCTS)**

8.37 SCR (SCR)—Offset 1Ch

Scratchpad Register

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7:0	0h RW	**scr (scr):** This register is for programmers to use as a temporary storage space. It has no defined purpose in the DW_apb_uart. Reset Value: 0x0

8.38 SRBR_STHR0 (SRBR_STHR0)—Offset 30h

Shadow Receive Buffer Register and Shadow Transmit Holding Register 0

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7:0	0h RW	**srbr_sthr0 (srbr_sthr0):** This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs. Reset Value: 0x0

8.39 SRBR_STHR1 (SRBR_STHR1)—Offset 34h

Shadow Receive Buffer Register and Shadow Transmit Holding Register 1

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7:0	0h RW	**srbr_sthr1 (srbr_sthr1):** This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs. Reset Value: 0x0

8.40 SRBR_STHR2 (SRBR_STHR2)—Offset 38h

Shadow Receive Buffer Register and Shadow Transmit Holding Register 2

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**srbr_sthr2 (srbr_sthr2):** This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs. Reset Value: 0x0

8.41 SRBR_STHR3 (SRBR_STHR3)—Offset 3Ch

Shadow Receive Buffer Register and Shadow Transmit Holding Register 3

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7:0	0h RW	**srbr_sthr3 (srbr_sthr3):** This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs. Reset Value: 0x0

8.42 SRBR_STHR4 (SRBR_STHR4)—Offset 40h

Shadow Receive Buffer Register and Shadow Transmit Holding Register 4

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7:0	0h RW	**srbr_sthr4 (srbr_sthr4):** This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs. Reset Value: 0x0

8.43 SRBR_STHR5 (SRBR_STHR5)—Offset 44h

Shadow Receive Buffer Register and Shadow Transmit Holding Register 5

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**srbr_sthr5 (srbr_sthr5):** This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs. Reset Value: 0x0

8.44 SRBR_STHR6 (SRBR_STHR6)—Offset 48h

Shadow Receive Buffer Register and Shadow Transmit Holding Register 6

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7:0	0h RW	**srbr_sthr6 (srbr_sthr6):** This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs. Reset Value: 0x0

8.45 SRBR_STHR7 (SRBR_STHR7)—Offset 4Ch

Shadow Receive Buffer Register and Shadow Transmit Holding Register 7

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7:0	0h RW	**srbr_sthr7 (srbr_sthr7):** This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs. Reset Value: 0x0

8.46 SRBR_STHR8 (SRBR_STHR8)—Offset 50h

Shadow Receive Buffer Register and Shadow Transmit Holding Register 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**srbr_sthr8 (srbr_sthr8):** This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs. Reset Value: 0x0

8.47 SRBR_STHR9 (SRBR_STHR9)—Offset 54h

Shadow Receive Buffer Register and Shadow Transmit Holding Register 9

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7:0	0h RW	**srbr_sthr9 (srbr_sthr9):** This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs. Reset Value: 0x0

8.48 SRBR_STHR10 (SRBR_STHR10)—Offset 58h

Shadow Receive Buffer Register and Shadow Transmit Holding Register 10

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7:0	0h RW	**srbr_sthr10 (srbr_sthr10):** This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs. Reset Value: 0x0

8.49 SRBR_STHR11 (SRBR_STHR11)—Offset 5Ch

Shadow Receive Buffer Register and Shadow Transmit Holding Register 11

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**srbr_sthr11 (srbr_sthr11):** This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs. Reset Value: 0x0

8.50 SRBR_STHR12 (SRBR_STHR12)—Offset 60h

Shadow Receive Buffer Register and Shadow Transmit Holding Register 12

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7:0	0h RW	**srbr_sthr12 (srbr_sthr12):** This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs. Reset Value: 0x0

8.51 SRBR_STHR13 (SRBR_STHR13)—Offset 64h

Shadow Receive Buffer Register and Shadow Transmit Holding Register 13

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7:0	0h RW	**srbr_sthr13 (srbr_sthr13):** This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs. Reset Value: 0x0

8.52 SRBR_STHR14 (SRBR_STHR14)—Offset 68h

Shadow Receive Buffer Register and Shadow Transmit Holding Register 14

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**srbr_sthr14 (srbr_sthr14):** This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs. Reset Value: 0x0

8.53 SRBR_STHR15 (SRBR_STHR15)—Offset 6Ch

Shadow Receive Buffer Register and Shadow Transmit Holding Register 15

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7:0	0h RW	**srbr_sthr15 (srbr_sthr15):** This is a shadow register for the RBR and has been allocated sixteen 32-bit locations so as to accommodate burst accesses from the master. This register contains the data byte received on the serial input port (sin) in UART mode or the serial infrared input (sir_in) in infrared mode. The data in this register is valid only if the Data Ready (DR) bit in the Line status Register (LSR) is set. If in non-FIFO mode (FIFO_MODE == NONE) or FIFOs are disabled (FCR[0] set to zero), the data in the RBR must be read before the next data arrives, otherwise it is overwritten, resulting in an overrun error. If in FIFO mode (FIFO_MODE != NONE) and FIFOs are enabled (FCR[0] set to one), this register accesses the head of the receive FIFO. If the receive FIFO is full and this register is not read before the next data character arrives, then the data already in the FIFO are preserved, but any incoming data is lost. An overrun error also occurs. Reset Value: 0x0

8.54 FAR (FAR)—Offset 70h

FIFO Access Register

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	Reserved.
0	0h RW	**srbr_sthr (srbr_sthr):** Writes have no effect when FIFO_ACCESS == No, always readable. This register is use to enable a FIFO access mode for testing, so that the receive FIFO can be written by the master and the transmit FIFO can be read by the master when FIFOs are implemented and enabled. When FIFOs are not implemented or not enabled it allows the RBR to be written by the master and the THR to be read by the master. 0 = FIFO access mode disabled 1 = FIFO access mode enabled Note, that when the FIFO access mode is enabled/disabled, the control portion of the receive FIFO and transmit FIFO is reset and the FIFOs are treated as empty. Reset Value: 0x0

8.55 TFR (TFR)—Offset 74h

Transmit FIFO Read

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7:0	0h RW	**tfr (tfr):** Transmit FIFO Read. These bits are only valid when FIFO access mode is enabled (FAR[0] is set to one). When FIFOs are implemented and enabled, reading this register gives the data at the top of the transmit FIFO. Each consecutive read pops the transmit FIFO and gives the next data value that is currently at the top of the FIFO. When FIFOs are not implemented or not enabled, reading this register gives the data in the THR. Reset Value: 0x0

8.56 RFW (RFW)—Offset 78h

Receive FIFO Write

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h NA	**Reserved (Res_31_10)**
9	0h WO	**RFFE (RFFE)**
8	0h WO	**RFPE (RFPE)**
7:0	0h WO	**RFWD (RFWD)**

8.57 USR (USR)—Offset 7Ch

UART Status Register

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 6h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RO	Reserved.
4	0h RO	**RFF (RFF):** Receive FIFO Full. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the receive FIFO is completely full. 0 = Receive FIFO not full 1 = Receive FIFO Full This bit is cleared when the RX FIFO is no longer full. Reset Value: 0x0
3	0h RO	**RFNE (RFNE):** Receive FIFO Not Empty. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the receive FIFO contains one or more entries. 0 = Receive FIFO is empty 1 = Receive FIFO is not empty This bit is cleared when the RX FIFO is empty. Reset Value: 0x0
2	1h RO	**TFE (TFE):** Transmit FIFO Empty. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the transmit FIFO is completely empty. 0 = Transmit FIFO is not empty 1 = Transmit FIFO is empty This bit is cleared when the TX FIFO is no longer empty. Reset Value: 0x1

Bit Range	Default & Access	Field Name (ID): Description
1	1h RO	**TFNF (TFNF):** Transmit FIFO Not Full. This bit is only valid when FIFO_STAT == YES. This is used to indicate that the transmit FIFO in not full. 0 = Transmit FIFO is full 1 = Transmit FIFO is not full This bit is cleared when the TX FIFO is full. Reset Value: 0x1
0	0h RO	**BUSY (BUSY):** UART Busy. This bit is valid only when UART_16550_COMPATIBLE == NO and indicates that a serial transfer is in progress, ; when cleared, indicates that the DW_apb_uart is idle or inactive. 0 = DW_apb_uart is idle or inactive 1 = DW_apb_uart is busy (actively transferring data) NOTE: It is possible for the UART Busy bit to be cleared even though a new character may have been sent from another device. That is, if the DW_apb_uart has no data in THR and RBR and there is no transmission in progress and a start bit of a new character has just reached the DW_apb_uart. This is due to the fact that a valid start is not seen until the middle of the bit period and this duration is dependent on the baud divisor that has been programmed. If a second system clock has been implemented (CLOCK_MODE == Enabled), the assertion of this bit is also delayed by several cycles of the slower clock. Reset Value: 0x0

8.58 TFL (TFL)—Offset 80h

Transmit FIFO Level

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RO	Reserved.
4:0	0h RO	**tfl (tfl):** Transmit FIFO Level. This is indicates the number of data entries in the transmit FIFO. Reset Value: 0x0

8.59 RFL (RFL)—Offset 84h

Receive FIFO Level

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RO	Reserved.
4:0	0h RO	**rfl (rfl):** Receive FIFO Level. This is indicates the number of data entries in the receive FIFO. Reset Value: 0x0

8.60 SRR (SRR)—Offset 88h

Software Reset Register

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RO	Reserved.
2	0h RW	**XFR (XFR):** XMIT FIFO Reset. This is a shadow register for the XMIT FIFO Reset bit (FCR[2]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the transmit FIFO. This resets the control portion of the transmit FIFO and treats the FIFO as empty. This also de-asserts the DMA TX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit. Reset Value: 0x0 Dependencies: Writes have no effect when FIFO_MODE == None.
1	0h RW	**RFR (RFR):** RCVR FIFO Reset. This is a shadow register for the RCVR FIFO Reset bit (FCR[1]). This can be used to remove the burden on software having to store previously written FCR values (which are pretty static) just to reset the receive FIFO This resets the control portion of the receive FIFO and treats the FIFO as empty. This also de-asserts the DMA RX request and single signals when additional DMA handshaking signals are selected (DMA_EXTRA == YES). Note that this bit is 'self-clearing'. It is not necessary to clear this bit. Reset Value: 0x0 Dependencies: Writes have no effect when FIFO_MODE == None.
0	0h RW	**UR (UR):** UART Reset. This asynchronously resets the DW_apb_uart and synchronously removes the reset assertion. For a two clock implementation both pclk and sclk domains are reset. Reset Value: 0x0

8.61 SRTS (SRTS)—Offset 8Ch

Shadow Request to Send

Access Method

Type: MEM Register (Size: 32 bits)	**Device:** **Function:**

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	Reserved.
0	0h RW	**srts (srts):** Shadow Request to Send. This is a shadow register for the RTS bit (MCR[1]), this can be used to remove the burden of having to performing a read-modify-write on the MCR. This is used to directly control the Request to Send (rts_n) output. The Request To Send (rts_n) output is used to inform the modem or data set that the DW_apb_uart is ready to exchange data. When Auto RTS Flow Control is not enabled (MCR[5] = 0), the rts_n signal is set low by programming MCR[1] (RTS) to a high. In Auto Flow Control, AFCE_MODE == Enabled and active (MCR[5] = 1) and FIFOs enable (FCR[0] = 1), the rts_n output is controlled in the same way, but is also gated with the receiver FIFO threshold trigger (rts_n is inactive high when above the threshold). Note that in Loopback mode (MCR[4] = 1), the rts_n output is held inactive-high while the value of this location is internally looped back to an input. Reset Value: 0x0

8.62 SBCR (SBCR)—Offset 90h

Shadow Break Control Bit. This is a shadow register for the Break bit (LCR[6]), this can be used to remove the burden of having to performing a read modify write on the LCR. This is used to cause a break condition to be transmitted to the receiving device. If set to one the serial output is forced to the spacing (logic 0) state. When not in Loopback Mode, as determined by MCR[4], the sout line is forced low until the Break bit is cleared. If SIR_MODE == Enabled and active (MCR[6] = 1) the sir_out_n line is continuously pulsed. When in Loopback Mode, the break condition is internally looped back to the receiver. Reset Value: 0x0

Access Method

Type: MEM Register (Size: 32 bits)	**Device:** **Function:**

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	Reserved.
0	0h RW	**sbcb (sbcb):** Shadow Break Control Register

8.63 SDMAM (SDMAM)—Offset 94h

Shadow DMA Mode

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	Reserved.
0	0h RW	**sdmam (sdmam):** Shadow DMA Mode. This is a shadow register for the DMA mode bit (FCR[3]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the DMA Mode bit gets updated. This determines the DMA signalling mode used for the dma_tx_req_n and dma_rx_req_n output signals when additional DMA handshaking signals are not selected (DMA_EXTRA == NO). 0 = mode 0 1 = mode 1 Reset Value: 0x0

8.64 SFE (SFE)—Offset 98h

Shadow FIFO Enable

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	Reserved.
0	0h RW	**sfe (sfe):** Shadow FIFO Enable. This is a shadow register for the FIFO enable bit (FCR[0]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the FIFO enable bit gets updated.This enables/disables the transmit (XMIT) and receive (RCVR) FIFOs. If this bit is set to zero (disabled) after being enabled then both the XMIT and RCVR controller portion of FIFOs are reset. Reset Value: 0x0

8.65 SRT (SRT)—Offset 9Ch

Shadow RCVR Trigger

Access Method

Type: MEM Register (Size: 32 bits)	**Device:** **Function:**

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	Reserved.
1:0	0h RW	**srt (srt):** Shadow RCVR Trigger. This is a shadow register for the RCVR trigger bits (FCR[7:6]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the RCVR trigger bit gets updated. This is used to select the trigger level in the receiver FIFO at which the Received Data Available Interrupt is generated. It also determines when the dma_rx_req_n signal is asserted when DMA Mode (FCR[3]) = 1. The following trigger levels are supported: 00 = 1 character in the FIFO 01 = FIFO full 10 = FIFO full 11 = FIFO 2 less than full Reset Value: 0x0

8.66 STET (STET)—Offset A0h

Shadow TX Empty Trigger. This is a shadow register for the TX empty trigger bits (FCR[5:4]). This can be used to remove the burden of having to store the previously written value to the FCR in memory and having to mask this value so that only the TX empty trigger bit gets updated. This is used to select the empty threshold level at which the THRE Interrupts are generated when the mode is active. The following trigger levels are supported: 00 = FIFO empty 01 = 2 characters in the FIFO 10 = FIFO full 11 = FIFO full Reset Value: 0x0 Dependencies: Writes have no effect when THRE_MODE_USER == Disabled.

Access Method

Type: MEM Register (Size: 32 bits)	**Device:** **Function:**

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	Reserved.
1:0	0h RW	**stet (stet):** Shadow TX Empty Trigger

8.67 HTX (HTX)—Offset A4h

Halt TX

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	Reserved.
0	0h RW	**htx (htx):** This register is use to halt transmissions for testing, so that the transmit FIFO can be filled by the master when FIFOs are implemented and enabled. 0 = Halt TX disabled 1 = Halt TX enabled Note, if FIFOs are implemented and not enabled, the setting of the halt TX register has no effect on operation. Reset Value: 0x0 Dependencies: Writes have no effect when FIFO_MODE == None.

8.68 DMASA (DMASA)—Offset A8h

DMA Software Acknowledge

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	Reserved.
0	0h WO	**dmasa (dmasa):** This register is use to perform a DMA software acknowledge if a transfer needs to be terminated due to an error condition. For example, if the DMA disables the channel, then the DW_apb_uart should clear its request. This causes the TX request, TX single, RX request and RX single signals to de-assert. Note that this bit is 'self-clearing'. It is not necessary to clear this bit. Reset Value: 0x0 Dependencies: Writes have no effect when DMA_EXTRA == No.

8.69 CPR (CPR)—Offset F4h

Component Parameter Register

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 43F32h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	Reserved.
23:16	4h RO	**FIFO_MODE (FIFO_MODE):** 0x00 = 0 0x01 = 16 0x02 = 32 to 0x80 = 2048 0x81- 0xff = reserved
15:14	0h RO	Reserved.
13	1h RO	**DMA_EXTRA (DMA_EXTRA):** 0 = FALSE, 1 = TRUE
12	1h RO	**UART_ADD_ENCODED_PARAMS (UART_ADD_ENCODED_PARAMS):** 0 = FALSE, 1 = TRUE
11	1h RO	**SHADOW (SHADOW):** 0 = FALSE, 1 = TRUE
10	1h RO	**FIFO_STAT (FIFO_STAT):** 0 = FALSE, 1 = TRUE
9	1h RO	**FIFO_ACCESS (FIFO_ACCESS):** 0 = FALSE, 1 = TRUE
8	1h RO	**ADDITIONAL_FEAT (ADDITIONAL_FEAT):** 0 = FALSE, 1 = TRUE
7	0h RO	**SIR_LP_MODE (SIR_LP_MODE):** 0 = FALSE, 1 = TRUE
6	0h RO	**SIR_MODE (SIR_MODE):** 0 = FALSE, 1 = TRUE
5	1h RO	**THRE_MODE (THRE_MODE):** 0 = FALSE, 1 = TRUE
4	1h RO	**AFCE_MODE (AFCE_MODE):** 0 = FALSE, 1 = TRUE
3:2	0h RO	Reserved.
1:0	2h RO	**APB_DATA_WIDTH (APB_DATA_WIDTH):** 00 = 8 bits 01 = 16 bits 10 = 32 bits 11 = reserved

8.70 UCV (UCV)—Offset F8h

UART Component Version

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 3331342Ah

Bit Range	Default & Access	Field Name (ID): Description
31:0	3331342Ah RO	**UART (UART):** ASCII value for each number in the version, followed by *. For example 32_30_31_2A represents the version 2.01* Reset Value: See the releases table in the AMBA 2 release notes.

8.71 CTR (CTR)—Offset FCh

Component Type Register

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 44570110h

Bit Range	Default & Access	Field Name (ID): Description
31:0	44570110h RO	**PeripheralID (PeripheralID):** This register contains the peripherals identification code. Reset Value: 0x44570110

8.72 RID - Bridge revision ID Register (RID)—Offset 30h

register contains the Bridge Revision ID

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved0:** Reserved
7:0	0h RO	**RID:** Revision ID of the Bridge.

8.73 GEN_REGRW1 - General Purpose register (GEN_REGRW1)—Offset 600h

General Purpose PRV RW Register 1.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW1:** This register value is brought out as oob_gen_prv_rw_reg1 out of band signal

8.74 GEN_REGRW2 - General Purpose register (GEN_REGRW2)—Offset 604h

General Purpose PRV RW Register 2.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW2:** This register value is brought out as oob_gen_prv_rw_reg2 out of band signal

8.75 GEN_REGRW3 - General Purpose register (GEN_REGRW3)—Offset 608h

General Purpose PRV RW Register 3.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW3:** This register value is brought out as oob_gen_prv_rw_reg3 out of band signal

8.76 GEN_REGRW4 - General Purpose register (GEN_REGRW4)—Offset 60Ch

General Purpose PRV RW Register 4.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW4:** This register value is brought out as oob_gen_prv_rw_reg4 out of band signal

8.77 Identifiers (ID)—Offset 0h

Access Method

Type: CFG Register (Size: 32 bits)
Device: 31
Function: 0

Default: 608086h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RO/V	**Device Identification 6 (DID_6):** Device Identification 6 is hardware reset to 9'b000000000. It can be overridden by the SetID IOSF-SB Message. Note: Refer to chapter 6s global DevID table for details.
22:21	3h RO/V	**Device Identification 5 (DID_5):** Device Identification 5 is hardwired to 2'b11.
20	0h RO/V	**Device Identification 4 (DID_4):** Device Identification 4 is set by the dtfus_core_lpcdid[4] fuse. Note: Refer to chapter 6s global DevID table for details.
19	0h RO/V	**Device Identification 3 (DID_3):** Device Identification 3 is set by the dtfus_core_lpcdid[3] fuse. Note: Refer to chapter 6s global DevID table for details.
18	0h RO/V	**Device Identification 2 (DID_2):** Device Identification 2 is set by the dtfus_core_lpcdid[2] fuse. Note: Refer to chapter 6s global DevID table for details.
17	0h RO/V	**Device Identification 1 (DID_1):** Device Identification 1 is set by the dtfus_core_lpcdid[1] fuse. Note: Refer to chapter 6s global DevID table for details.
16	0h RO/V	**Device Identification 0 (DID_0):** Device Identification 0 is set by the dtfus_core_lpcdid[0] fuse. Note: Refer to chapter 6s global DevID table for details.
15:0	8086h RO	**Vendor Identification (VID):** Indicates Intel.

8.78 Device Command (CMD)—Offset 4h

Access Method

Type: CFG Register (Size: 16 bits)
Device: 31
Function: 0

Default: 7h

Bit Range	Default & Access	Field Name (ID): Description
15:11	0h RO	**Reserved (RSVD):** Reserved.
10	0h RO	**Interrupt Disable (ID):** The LPC bridge has no interrupts to disable.
9	0h RO	**Fast Back to Back Enable (FBE):** Reserved as 0 per PCI-Express spec.
8	0h RW	**SERR# Enable (SEE):** The LPC bridge generates SERR# if this bit is set.
7	0h RO	**Wait Cycle Control (WCC):** Reserved as 0 per PCI-Express spec.
6	0h RW	**Parity Error Response Enable (PERE):** When this bit is set to 1, it enables the LPC bridge to response to parity errors detected on backbone interface.
5	0h RO	**VGA Palette Snoop (VGA_PSE):** Reserved as 0 per PCI-Express spec.
4	0h RO	**Memory Write and Invalidate Enable (MWIE):** Reserved as 0 per PCI-Express spec.
3	0h RO	**Special Cycle Enable (SCE):** Reserved as 0 per PCI-Express spec.
2	1h RO	**Bus Master Enable (BME):** Bus Masters cannot be disabled.
1	1h RO	**Memory Space Enable (MSE):** Memory space cannot be disabled on LPC.
0	1h RO	**I/O Space Enable (IOSE):** I/O space cannot be disabled on LPC.

8.79 Status (STS)—Offset 6h

Access Method

Type: CFG Register (Size: 16 bits)
Device: 31
Function: 0

Default: 200h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**Detected Parity Error (DPE):** Set when the bridge detects a parity error on the internal backbone. This bit gets set even if CMD.PERE is not set.
14	0h RW/1C	**Signaled System Error (SSE):** Set when the LPC bridge signals a system error to the internal SERR# logic.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RO	**Received Master Abort (RMA):** Set when the bridge receives a completion with unsupported request status from the backbone. LPC is a target only controller.
12	0h RO	**Received Target Abort (RTA):** Set when the bridge receives a completion with completer abort status from the backbone. LPC is a target only controller.
11	0h RW/1C	**Signalled Target Abort (STA):** Set when the bridge generates a completion packet with target abort status on the backbone.
10:9	1h RO	**DEVSEL# Timing Status (DTS):** Indicates medium timing, although this has no meaning on the backbone.
8	0h RW/1C	**Data Parity Error Detected (DPD):** Set when the bridge receives a completion packet from the backbone from a previous request, and detects a parity error, and CMD.PERE is set.
7	0h RO	**Fast Back to Back Capable (FBC):** Reserved - bit has no meaning on internal backbone.
6	0h RO	**Reserved (RSVD):** Reserved.
5	0h RO	**66 MHz Capable (C66):** Reserved - bit has no meaning on internal backbone.
4	0h RO	**Capabilities List (CLIST):** There is a capabilities list in the LPC bridge.
3	0h RO	**Interrupt Status (IS):** The LPC bridge does not generate interrupts.
2:0	0h RO	**Reserved (RSVD_1):** Reserved.

8.80 Revision ID (RID)—Offset 8h

Access Method

Type: CFG Register
(Size: 8 bits)

Device: 31
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO/V	**Revision ID (RID):** Indicates the part revision. This will reset to 0 but is expected to be overridden by the SetID IOSF-SB message.

8.81 Class Code (CC)—Offset 9h

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 31
Function: 0

Default: 60100h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	Reserved.
23:16	6h RO	**Base Class Code (BCC):** Indicates the device is a bridge device.
15:8	1h RO	**Sub-Class Code (SCC):** Indicates the device a PCI to ISA bridge.
7:0	0h RO	**Programming Interface (PI):** The LPC bridge has no programming interface.

8.82 Primary Latency Timer (PLT)—Offset Dh

Access Method

Type: CFG Register
(Size: 8 bits)

Device: 31
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:3	0h RO	**Master Latency Count (MLC):** Reserved per 3GIO spec.
2:0	0h RO	**Reserved (RSVD):** Reserved.

8.83 Header Type (HTYPE)—Offset Eh

Access Method

Type: CFG Register
(Size: 8 bits)

Device: 31
Function: 0

Default: 80h

Bit Range	Default & Access	Field Name (ID): Description
7	1h RO	**Multi-function Device (MFD):** This bit is 1 to indicate a multifunction device.
6:0	0h RO	**Header Type (HTYPE):** Identifies the header layout of the configuration space, which is a generic device.

8.84 Sub System Identifiers (SS)—Offset 2Ch

This register is initialized to logic 0 by the assertion of PLTRST#. This register can be written only once after PLTRST# de-assertion.

Access Method

Type: CFG Register (Size: 32 bits)
Device: 31
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**Subsystem ID (SSID):** This is written by BIOS. No hardware action taken on this value.
15:0	0h RW/O	**Subsystem Vendor ID (SSVID):** This is written by BIOS. No hardware action taken on this value.

8.85 Capability List Pointer (CAPP)—Offset 34h

Access Method

Type: CFG Register (Size: 8 bits)
Device: 31
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Capability Pointer (CP):** Indicates the offset of the first Capability Item.

8.86 Serial IRQ Control (SCNT)—Offset 64h

Access Method

Type: CFG Register (Size: 8 bits)
Device: 31
Function: 0

Default: 10h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**Enable (EN):** When set, serial IRQs will be recognized.
6	0h RW	**Mode (MD):** When set, the serial IRQ machine will be in continuous mode. When cleared, the serial IRQ machine will be in quiet mode. When setting the EN bit, this bit must also be written as a one to guarantee that the first action of the serial IRQ machine will be a start frame.

Bit Range	Default & Access	Field Name (ID): Description
5:2	4h RO	**Frame Size (FS):** Fixed field that indicates the size of the SERIRQ frame as 21 frames.
1:0	0h RW	**Start Frame Pulse Width (SFPW):** This is the number of 33 MHz clocks that the SERIRQ pin will be driven low by the Serial IRQ controller to signal a start frame. In continuous mode, the controller will drive the start frame for the number of clocks specified. In quiet mode, the controller will drive the start frame for the number of clocks specified minus one, as the first clock was driven by the peripheral. Bits Clocks • 00: 4 • 01: 6 • 10: 8 • 11: Reserved

8.87 I/O Decode Ranges (IOD)—Offset 80h

Access Method

Type: CFG Register (Size: 16 bits)
Device: 31
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	**Reserved (RSVD):** Reserved.
12	0h RW	**FDD Range (FDD):** The following table describes which range to decode for the FDD Port. Bits Decode Range • 0 3F0h - 3F5h, 3F7h (Primary) • 1 370h - 375h, 377h (Secondary)
11:10	0h RO	**Reserved (RSVD_1):** Reserved.
9:8	0h RW	**LPT Range (LPT):** The following table describes which range to decode for the LPT Port: Bits Decode Range • 00 378h - 37Fh and 778h - 77Fh • 01 278h - 27Fh (port 279h is read only) and 678h - 67Fh • 10 3BCh - 3BEh and 7BCh - 7BEh • 11 Reserved
7	0h RO	**Reserved (RSVD_2):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
6:4	0h RW	**ComB Range (CB):** The following table describes which range to decode for the COMB Port. Bits Decode Range • 000 3F8h - 3FFh (COM 1) • 001 2F8h - 2FFh (COM 2) • 010 220h - 227h • 011 228h - 22Fh • 100 238h - 23Fh • 101 2E8h - 2EFh (COM 4) • 110 338h - 33Fh • 111 3E8h - 3EFh (COM 3)
3	0h RO	**Reserved (RSVD_3):** Reserved.
2:0	0h RW	**ComA Range (CA):** The following table describes which range to decode for the COMA Port. Bits Decode Range • 000 3F8h - 3FFh (COM 1) • 001 2F8h - 2FFh (COM 2) • 010 220h - 227h • 011 228h - 22Fh • 100 238h - 23Fh • 101 2E8h - 2EFh (COM 4) • 110 338h - 33Fh • 111 3E8h - 3EFh (COM 3)

8.88 I/O Enables (IOE)—Offset 82h

Access Method

Type: CFG Register
(Size: 16 bits)

Device: 31
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RO	**Reserved (RSVD):** Reserved.
13	0h RW	**Microcontroller Enable #2 (ME2):** Enables decoding of I/O locations 4Eh and 4Fh to LPC.
12	0h RW	**SuperI/O Enable (SE):** Enables decoding of I/O locations 2Eh and 2Fh to LPC.
11	0h RW	**Microcontroller Enable #1 (ME1):** Enables decoding of I/O locations 62h and 66h to LPC.
10	0h RW	**Keyboard Enable (KE):** Enables decoding of the keyboard I/O locations 60h and 64h to LPC.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**High Gameport Enable (HGE):** Enables decoding of the I/O locations 208h to 20Fh to LPC.
8	0h RW	**Low Gameport Enable (LGE):** Enables decoding of the I/O locations 200h to 207h to LPC.
7:4	0h RO	**Reserved (RSVD_1):** Reserved
3	0h RW	**Floppy Drive Enable (FDE):** Enables decoding of the FDD range to LPC. Range is selected by LIOD.FDE
2	0h RW	**Parallel Port Enable (PPE):** Enables decoding of the LPT range to LPC. Range is selected by LIOD.LPT.
1	0h RW	**Com Port B Enable (CBE):** Enables decoding of the COMB range to LPC. Range is selected LIOD.CB.
0	0h RW	**Com Port A Enable (CAE):** Enables decoding of the COMA range to LPC. Range is selected LIOD.CA.

8.89 LPC Generic I/O Range #1 (LGIR1)—Offset 84h

Access Method

Type: CFG Register (Size: 32 bits)
Device: 31
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved.
23:18	0h RW	**Address[7:2] Mask (Address_7_2_Mask):** A 1 in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match. The corresponding bit in the Address field, below, is ignored. The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size.
17:16	0h RO	**Reserved (RSVD_1):** Reserved.
15:2	0h RW	**Address[15:2] (Address_15_2):** DWord-aligned address. Note that PCH does not provide decode down to the word or byte level.
1	0h RO	**Reserved (RSVD_2):** Reserved.
0	0h RW	**LPC Decode Enable (LPC_Decode_Enable):** When this bit is set to 1, then the range specified in this register is enabled for decoding to LPC.

8.90 LPC Generic I/O Range #2 (LGIR2)—Offset 88h

Access Method

Type: CFG Register (Size: 32 bits)
Device: 31
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved.
23:18	0h RW	**Address[7:2] Mask (Address_7_2_Mask):** A 1 in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match. The corresponding bit in the Address field, below, is ignored. The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size.
17:16	0h RO	**Reserved (RSVD_1):** Reserved.
15:2	0h RW	**Address[15:2] (Address_15_2):** DWord-aligned address. Note that PCH does not provide decode down to the word or byte level.
1	0h RO	**Reserved (RSVD_2):** Reserved.
0	0h RW	**LPC Decode Enable (LPC_Decode_Enable):** When this bit is set to 1, then the range specified in this register is enabled for decoding to LPC.

8.91 LPC Generic I/O Range #3 (LGIR3)—Offset 8Ch

Access Method

Type: CFG Register (Size: 32 bits)
Device: 31
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved.
23:18	0h RW	**Address[7:2] Mask (Address_7_2_Mask):** A 1 in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match. The corresponding bit in the Address field, below, is ignored. The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size.
17:16	0h RO	**Reserved (RSVD_1):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
15:2	0h RW	**Address[15:2] (Address_15_2):** DWord-aligned address. Note that PCH does not provide decode down to the word or byte level.
1	0h RO	**Reserved (RSVD_2):** Reserved.
0	0h RW	**LPC Decode Enable (LPC_Decode_Enable):** When this bit is set to 1, then the range specified in this register is enabled for decoding to LPC.

8.92 LPC Generic I/O Range #4 (LGIR4)—Offset 90h

Access Method

Type: CFG Register (Size: 32 bits)
Device: 31
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved.
23:18	0h RW	**Address[7:2] Mask (Address_7_2_Mask):** A 1 in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match. The corresponding bit in the Address field, below, is ignored. The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size.
17:16	0h RO	**Reserved (RSVD_1):** Reserved.
15:2	0h RW	**Address[15:2] (Address_15_2):** DWord-aligned address. Note that PCH does not provide decode down to the word or byte level.
1	0h RO	**Reserved (RSVD_2):** Reserved.
0	0h RW	**LPC Decode Enable (LPC_Decode_Enable):** When this bit is set to 1, then the range specified in this register is enabled for decoding to LPC.

8.93 USB Legacy Keyboard/Mouse Control (ULKMC)—Offset 94h

Access Method

Type: CFG Register (Size: 32 bits)
Device: 31
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved.
15	0h RW/1C	**SMI Caused by End of Pass-through (SMIBYENDPS):** Indicates if the event occurred. Note that even if the corresponding enable bit is not set in the Bit 7, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Writing a 1 to this bit will clear the latch.
14	0h RO	**Reserved (RSVD_1):** Reserved.
13	0h RW	**Reserved (RSVD_2):** Reserved.
12	0h RO	**Reserved (RSVD_3):** Reserved.
11	0h RW/1C	**SMI Caused by Port 64 Write (TRAPBY64W):** Indicates if the event occurred. Note that even if the corresponding enable bit is not set in the Bit 3, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Writing a 1 to this bit will clear the latch. Note that the A20Gate Pass-Through Logic allows specific port 64h Writes to complete without setting this bit.
10	0h RW/1C	**SMI Caused by Port 64 Read (TRAPBY64R):** Indicates if the event occurred. Note that even if the corresponding enable bit is not set in the Bit 2, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Writing a 1 to this bit will clear the latch.
9	0h RW/1C	**SMI Caused by Port 60 Write (TRAPBY60W):** Indicates if the event occurred. Note that even if the corresponding enable bit is not set in the Bit 1, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Writing a 1 to this bit will clear the latch. Note that the A20Gate Pass-Through Logic allows specific port 60h Writes to complete without setting this bit.
8	0h RW/1C	**SMI Caused by Port 60 Read (TRAPBY60R):** Indicates if the event occurred. Note that even if the corresponding enable bit is not set in the Bit 0, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Writing a 1 to this bit will clear the latch.
7	0h RW	**SMI at End of Pass-through Enable (SMIATENDPS):** May need to cause SMI at the end of a pass-through. Can occur if an SMI is generated in the middle of a pass through, and needs to be serviced later.
6	0h RO	**Pass Through State (PSTATE):** This read-only bit indicates that the state machine is in the middle of an A20GATE pass-through sequence. If software needs to reset this bit, it should set Bit 5 0.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**A20Gate Pass-Through Enable (A20PASSEN):** When enabled, allows A20GATE sequence Pass-Through function. When enabled, a specific cycle sequence involving writes to port 60h and port 64h does not result in the setting of the SMI status bits.SMI# will not be generated, even if the various enable bits are set.
4	0h RW	**Reserved (RSVD_4):** Reserved.
3	0h RW	**SMI on Port 64 Writes Enable (s64WEN):** When set, a 1 in bit 11 will cause an SMI event.
2	0h RW	**SMI on Port 64 Reads Enable (s64REN):** When set, a 1 in bit 10 will cause an SMI event.
1	0h RW	**SMI on Port 60 Writes Enable (s60WEN):** When set, a 1 in bit 9 will cause an SMI event.
0	0h RW	**SMI on Port 60 Reads Enable (s60REN):** When set, a 1 in bit 8 will cause an SMI event.

8.94 LPC Generic Memory Range (LGMR)—Offset 98h

Access Method

Type: CFG Register (Size: 32 bits)
Device: 31
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Memory Address[31:16] (MA_31_16):** This field specifies a 64KB memory block anywhere in the 4GB memory space that will be decoded to LPC as standard LPC Memory Cycle if enabled.
15:1	0h RO	**Reserved (RSVD):** Reserved.
0	0h RW	**LPC Memory Range Decode Enable (LMRD_En):** When this bit is set to 1, then the range specified in this register is enabled for decoding to LPC.

8.95 Reserved (RSVD)—Offset 9Ch

Reserved.

Access Method

Type: CFG Register (Size: 32 bits)
Device: 31
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD):** Reserved.

8.96 FWH ID Select #1 (FS1)—Offset D0h

This register contains the IDSEL fields the LPC Bridge uses for memory cycles going to the FWH.

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 31
Function: 0

Default: 112233h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**F8-FF IDSEL (IF8):** IDSEL to use in FWH cycle for range enabled by BDE.EF8.
27:24	0h RW	**F0-F7 IDSEL (IF0):** IDSEL to use in FWH cycle for range enabled by BDE.EF0.
23:20	1h RW	**E8-EF IDSEL (IE8):** IDSEL to use in FWH cycle for range enabled by BDE.EE8.
19:16	1h RW	**E0-E7 IDSEL (IE0):** IDSEL to use in FWH cycle for range enabled by BDE.EE0.
15:12	2h RW	**D8-DF IDSEL (ID8):** IDSEL to use in FWH cycle for range enabled by BDE.ED8.
11:8	2h RW	**D0-D7 IDSEL (ID0):** IDSEL to use in FWH cycle for range enabled by BDE.ED0.
7:4	3h RW	**C8-CF IDSEL (IC8):** IDSEL to use in FWH cycle for range enabled by BDE.EC8.
3:0	3h RW	**C0-C7 IDSEL (IC0):** IDSEL to use in FWH cycle for range enabled by BDE.EC0.

8.97 FWH ID Select #2 (FS2)—Offset D4h

This register contains the additional IDSEL fields the LPC Bridge uses for memory cycles going to the FWH.

Access Method

Type: CFG Register
(Size: 16 bits)

Device: 31
Function: 0

Default: 4567h

Bit Range	Default & Access	Field Name (ID): Description
15:12	4h RW	**70-7F IDSEL (I70):** IDSEL to use in FWH cycle for range enabled by BDE.E70.
11:8	5h RW	**60-6F IDSEL (I60):** IDSEL to use in FWH cycle for range enabled by BDE.E60.
7:4	6h RW	**50-5F IDSEL (I50):** IDSEL to use in FWH cycle for range enabled by BDE.E50.
3:0	7h RW	**40-4F IDSEL (I40):** IDSEL to use in FWH cycle for range enabled by BDE.E40.

8.98 BIOS Decode Enable (BDE)—Offset D8h

Note that this register effects the BIOS decode regardless of whether the BIOS is resident on LPC or SPI. The concept of Feature Space does not apply to SPI-based flash. PCH simply decodes these ranges as memory accesses when enabled for the SPI flash interface.

Access Method

Type: CFG Register (Size: 16 bits)
Device: 31
Function: 0

Default: FFCFh

Bit Range	Default & Access	Field Name (ID): Description
15	1h RO	**F8-FF Enable (EF8):** Enables decoding of 512K of the following: BIOS range: • Data space: FFF80000h - FFFFFFFFh • Feature space: FFB80000h - FFBFFFFFh
14	1h RW	**F0-F8 Enable (EF0):** Enables decoding of 512K of the following: BIOS range: • Data space: FFF00000h - FFF7FFFFh • Feature space: FFB00000h - FFB7FFFFh
13	1h RW	**E8-EF Enable (EE8):** Enables decoding of 512K of the following: BIOS range: • Data space: FFE80000h - FFEFFFFFh • Feature space: FFA80000h - FFAFFFFFh
12	1h RW	**E0-E8 Enable (EE0):** Enables decoding of 512K of the following: BIOS range: • Data space: FFE00000h - FFE7FFFFh • Feature Space: FFA00000h - FFA7FFFFh
11	1h RW	**D8-DF Enable (ED8):** Enables decoding of 512K of the following: BIOS range: • Data space: FFD80000h - FFDFFFFFh • Feature space: FF980000h - FF9FFFFFh
10	1h RW	**D0-D7 Enable (ED0):** Enables decoding of 512K of the following: BIOS range: • Data space: FFD00000h - FFD7FFFFh • Feature space: FF900000h - FF97FFFFh

Bit Range	Default & Access	Field Name (ID): Description
9	1h RW	**C8-CF Enable (EC8):** Enables decoding of 512K of the following: BIOS range: • Data space: FFC80000h - FFCFFFFFh • Feature space: FF880000h - FF8FFFFFh
8	1h RW	**C0-C7 Enable (EC0):** Enables decoding of 512K of the following: BIOS range: • Data space: FFC00000h - FFC7FFFFh • Feature space: FF800000h - FF87FFFFh
7	1h RW	**Legacy F Segment Enable (LFE):** This enables the decoding of the legacy 64KB range at F0000h - FFFFFh. Note that decode for the BIOS legacy F segment is enabled by the LFE bit only, it is not affected by the GEN_PMCON_1.iA64_EN bit.
6	1h RW	**Legacy E Segment Enable (LEE):** This enables the decoding of the legacy 64KB range at E0000h - EFFFFh. Note that decode for the BIOS legacy E segment is enabled by the LEE bit only, it is not affected by the GEN_PMCON_1.iA64_EN bit.
5:4	0h RO	**Reserved (RSVD):** Reserved.
3	1h RW	**70-7F Enable (E70):** Enables decoding of 1MB of the following: BIOS range: • Data space: FF700000h - FF7FFFFFh • Feature space: FF300000h - FF3FFFFFh
2	1h RW	**60-6F Enable (E60):** Enables decoding of 1MB of the following: BIOS range: • Data space: FF600000h - FF6FFFFFh • Feature Space: FF200000h - FF2FFFFFh
1	1h RW	**50-5F Enable (E50):** Enables decoding of 1MB of the following: BIOS range: • Data space: FF500000h - FF5FFFFFh • Feature space: FF100000h - FF1FFFFFh
0	1h RW	**40-4F Enable (E40):** Enables decoding of 1MB of the following: BIOS range: • Data space: FF400000h - FF4FFFFFh • Feature space: FF000000h - FF0FFFFFh

8.99 BIOS Control (BC)—Offset DCh

Access Method

Type: CFG Register
(Size: 8 bits)

Device: 31
Function: 0

Default: 20h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW/1L	**BIOS Interface Lock-Down (BILD):** When set, prevents BC.TS and BC.BBS from being changed. This bit can only be written from 0 to 1 once. BIOS Note: This bit is not backed up in the RTC well. This bit should also be set in the BUC register in the RTC device to record the last state of this value following a cold reset.
6	0h RW/L	**Boot BIOS Strap (BBS):** This field determines the destination of accesses to the BIOS memory range. Bits Description • 0: SPI • 1: LPC When SPI or LPC is selected, the range that is decoded is further qualified by other configuration bits described in the respective sections. The value in this field can be overwritten by software as long as the BIOS Interface Lock-Down is not set.
5	1h RW/L	**Enable InSMM.STS (EISS):** When this bit is set, the BIOS region is not writable until SMM sets the InSMM.STS bit. Today BIOS Flash is writable if WPD is a 1. If this bit [5] is set, then WPD must be a 1 and InSMM.STS (0xFED3_0880[0]) must be 1 also. If this bit [5] is clear, then BIOS is writable based only on WPD = 1 and the InSMM.STS is a don't care.
4	0h RO	**Top Swap (TS):** When set, PCH will invert either A16, A17, or A18 for cycles going to the BIOS space (but not the feature space) in the FWH. When cleared, PCH will not invert A16. If booting from LPC (FWH), then the Boot Block size is 64KB and A16 is inverted if Top Swap is enabled. If booting from SPI, then the BOOT_BLOCK_SIZE soft strap determines if A16, A17, or A18 should be inverted if Top Swap is enabled. *If PCH is strapped for Top-Swap (GNT[3]# is low at rising edge of PWROK), then this bit cannot be cleared by software. The strap jumper should be removed and the system rebooted. BIOS Note: 1. This bit provides a read-only path to view the state of the Top Swap strap. It is backed up and driven from the RTC well. Bios will need to program the corresponding register in the RTC Controller (in RTC well), which will be reflected in this register. 2. The Register portion of the Top Swap is lockable by the Bios Interface Lockdown Bit (BC.BILD) but unlockable by SPI Flash Protected Range and Top Swap Override (uCode.PRR_TS_OVR).

Bit Range	Default & Access	Field Name (ID): Description
3:2	0h RW	**SPI Read Configuration (SRC):** This 2-bit field controls two policies related to BIOS reads on the SPI interface: 1. Bit 3: Prefetch Enable 2. Bit 2: Cache Disable Settings are summarized below: • Bits 3:2 Description • 00: No prefetching, but caching enabled. Direct Memory reads load the read buffer cache with valid data, allowing repeated reads to the same range to complete quickly. • 01: No prefetching and no caching. One-to-one correspondence of host BIOS reads to SPI cycles. This value can be used to invalidate the cache. • 10: Prefetching and Caching enabled. This mode is used for long sequences of short reads to consecutive addresses (i.e. shadowing) • 11: Illegal. Caching must be enabled when Prefetching is enabled. This eliminates the need for a complex prefetch-flushing mechanism.
1	0h RW/1L	**Lock Enable (LE):** When set, setting the WP bit will cause SMI. When cleared, setting the WP bit will not cause SMI. Once set, this bit can only be cleared by a PLTRST#. When this bit is set, EISS - bit [5] of this register is locked down.
0	0h RW	**Write Protect Disable (WPD):** When set, access to the BIOS space is enabled for both read and write cycles to BIOS. When cleared, only read cycles are permitted to the FWH or SPI flash. When this bit is written from a 0 to a 1 and the LE bit is also set, an SMI# is generated. This ensures that only SMM code can update BIOS.

8.100 PCI Clock Control (PCCTL)—Offset E0h

Access Method

Type: CFG Register (Size: 32 bits)
Device: 31
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RO	**Reserved (RSVD):** Reserved.
9	0h RO/V	**CLKRUN# Buffer Enable Override (CLKRUN_EN_OVR):** When set to '1', SW is in control of the CLKRUN# buffer enable and the value in CLKRUN_EN_VAL will be propagated to the output buffer enable. When this bit is '0', HW will determine the value of the buffer enable.

Bit Range	Default & Access	Field Name (ID): Description
8	0h RO/V	**CLKRUN# Override (CLKRUN_OVR):** When set to '1', SW is in control of the CLKRUN# pin and the value in CLKRUN_VAL will be propagated to the output pin. When this bit is '0', HW will determine the value of the pin.
7	0h RO/V	**CLKRUN# Buffer Enable Value (CLKRUN_EN_VAL):** Either HW or SW may own control of the CLKRUN# pin. This bit provides the value to drive on the active low CLKRUN# buffer enable if CLKRUN_EN_OVR is set to '1'.
6	0h RO/V	**CLKRUN# Pin Output Value (CLKRUN_VAL):** Either HW or SW may own control of the CLKRUN# pin. This bit provides the value to drive on the pin if CLKRUN_OVR is set to '1'.
5	0h RO/V	**Stop PCI# Value (STP_PCI_VAL):** Either Hardware or Software may own control of the internal STP_PCI#. This bit provides the value to drive on the STP_PCI# if STP_PCI_OVR is set to 1. Note: 1) SW cannot control the STP_PCI# pin while PLTRST# is asserted (the pin will be at its reset default value).
4	0h RO/V	**Stop PCI# Override (STP_PCI_OVR):** When set to 1, Firmware is in control of the STP_PCI# and the value in STP_PCI_VAL will be propagated to the internal STP_PCI#. When this bit is '0', HW will determine the value of the pin. Note: 1) Bios cannot control the STP_PCI# pin while PLTRST# is asserted (the pin will be at its reset default value).
3:2	0h RW	**PCI Clock Valid Configuration (PCLKVLD_CFG):** This field determines the relationship between the internally broadcast indication of the external PCI clock being valid vs. the STP_PCI# pin. Encodings: • 00: 1 flop stage of delay from STP_PCI# (Default) - Expected setting for Buffer Through or Full Integration Mode • 01: No delay (edges match STP_PCI#) - For CK505 legacy mode • 10: 2 flop stages of delay from STP_PCI# - Risk mitigation • 11: Tie high (indicate that PCI clock is always valid) - Risk mitigation
1	0h RW	**Reserved (RSVD_1):** Reserved.
0	0h RW	**Clock Run Enable (CLKRUN_EN):** Enables the CLKRUN# logic to stop the PCI clocks. If the SLP_EN bit is set, then the Intel PCH will drive CLKRUN# low. This will keep the PCI and LPC clocks running on the way to the sleeping state. This is required to meet an LPC specification. This does not necessarily mean that the CLKRUN_EN bit is forced low when SLP_EN is set. Even though the CLKRUN# signal will be low when SLP_EN is set, the state of the CLKRUN_EN bit is ignored when SLP_EN bit is set. This gives flexibility in the implementation.

8.101 Manufacturer's ID (MANID)—Offset F8h

This register is assigned during the boot flow using the SetID message based on values found in the fuse block. All fields except MID will be reset by hardware to zero and set only by the SetID message. See Chapter 33 (DFT) for more information on the values that will be assigned in the fuse block for this component.

Access Method

Type: CFG Register (Size: 32 bits)
Device: 31
Function: 0

Default: F00h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD):** Reserved.
27:24	0h RO/V	**Dot portion of Process ID: (PD):** Indicates the dot process
23:16	0h RO/V	**Manufacturing Stepping Identifier (MSID):** This field is incremented for each stepping of the part. Note that this field can be used by software to differentiate steppings when the Stepping Revision ID may not change.
15:8	Fh RO/V	**Manufacturing Identifier (MID):** 0Fh = Intel
7:0	0h RO/V	**Process portion of process ID (PID):** Indicates the process. The dot portion of the process is reflected in bits [27:24].

§ §

9 eSPI Registers

This chapter documents the registers in Bus: 0, Device 31, Function 0.

9.1 Identifiers (ESPI_DID_VID) — Offset 0h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 0h	31978086 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	3197h RO/V	**Device Identification (DID):** The default value of this is hardwired. The upper 8-bits of this field can be overridden by the SetID IOSF-SB Message. Note:Refer to chapter 6s global DevID table for details .
15:0	8086h RO	**Vendor Identification (VID):** Indicates Intel

9.2 Device Status and Command (ESPI_STS_CMD) — Offset 4h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 4h	403 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C/V	**Detected Parity Error (DPE):** Set when the bridge detects a parity error on the internal backbone. This bit gets set even if CMD.PERE is not set.
30	0h RW/1C/V	**Signaled System Error (SSE):** Set when the LPC bridge signals a system error to the internal SERR# logic.
29	0h RW/1C/V	**Received Master Abort (RMA):** Set when the bridge receives a completion with unsupported request status from the backbone.
28	0h RW/1C/V	**Received Target Abort (RTA):** Set when the bridge receives a completion with completer abort status from the backbone.
27	0h RW/1C/V	**Signaled Target Abort (STA):** Set when the bridge generates a completion packet with target abort status on the backbone.
26:25	0h RO	**DEVSEL# Timing Status (DTS):** Indicates medium timing, although this has no meaning on the backbone.
24	0h RW/1C/V	**Data Parity Error Detected (DPD):** Set when the bridge receives a completion packet from the backbone from a previous request, and detects a parity error, and CMD.PERE is set.
23	0h RO	**Fast Back to Back Capable (FBC):** Reserved - bit has no meaning on internal backbone.
22	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
21	0h RO	**66 MHz Capable (C66):** Reserved - bit has no meaning on internal backbone.
20	0h RO	**Capabilities List (CLIST):** There is a capabilities list in the LPC bridge.
19	0h RO	**Interrupt Status (IS):** The LPC bridge does not generate interrupts.
18:11	0h RO	**Reserved (RSVD):** Reserved
10	1h RO	**Interrupt Disable (ID):** The LPC bridge has no interrupts to disable
9	0h RO	**Fast Back to Back Enable (FBE):** Reserved as 0 per PCI-Express spec.
8	0h RW	**SERR# Enable (SEE):** The LPC bridge generates SERR# if this bit is set.
7	0h RO	**Wait Cycle Control (WCC):** Reserved as 0 per PCI-Express spec.
6	0h RW	**Parity Error Response Enable (PERE):** When this bit is set to 1, it enables the LPC bridge to response to parity errors detected on backbone interface.
5	0h RO	**VGA Palette Snoop (VGA_PSE):** Reserved as 0 per PCI-Express spec.
4	0h RO	**Memory Write and Invalidate Enable (MWIE):** Reserved as 0 per PCI-Express spec.
3	0h RO	**Special Cycle Enable (SCE):** Reserved as 0 per PCI-Express spec.
2	0h RW	**Bus Master Enable (BME):** Bus Masters cannot be disabled.
1	1h RO	**Memory Space Enable (MSE):** Memory space cannot be disabled on LPC.
0	1h RO	**I/O Space Enable (IOSE):** I/O space cannot be disabled on LPC.

9.3 Class Code and Revision ID (ESPI_CC_RID) — Offset 8h

Writing to this register controls what is reported in all of the RID registers in the component. The value written does not get directly loaded in this register. However, the value is checked to determine which value to report. Once written, additional writes to this register must not have any effect until a core-well reset occurs. BIOS must always write to this register in order to guarantee that the functionality is locked.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 8h	6010000 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	6h RO	**Base Class Code (BCC):** Indicates the device is a bridge device.

Bit Range	Default & Access	Field Name (ID): Description
23:16	1h RO	**Sub-Class Code (SCC):** Indicates the device a PCI to ISA bridge.
15:8	0h RO	**Programming Interface (PI):** The LPC bridge has no programming interface.
7:0	0h RO/V	**Revision ID (RID):** Indicates the part revision This will reset to 0 but is expected to be overridden by the SetID IOSF-SB message.

9.4 Peripheral Channel Header Type and Primary Latency (ESPI_PCHT_PCPLT) — Offset Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + Ch	800000 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23	1h RO	**Multi-function Device (MFD):** This bit is 1 to indicate that this PCI function is part of a multi-function device.
22:16	0h RO	**Header Type (HTYPE):** Identifies the header layout of the configuration space, which is a generic device.
15:11	0h RO	**Master Latency Count (MLC):** Reserved
10:0	0h RO	**Reserved (RSVD):** Reserved

9.5 CSXE BAR0 MMIO (CSXE_ESPI_MBAR) — Offset 10h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 10h	8 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**Memory BAR (MEMBAR):** Software programs this register with the base address of the device's memory region
11:4	0h RO	**Memory Size (MEMSIZE):** Hard wired to 0 to indicate 4KB of memory space
3	1h RO	**PREFETCH:** Set to '1' to indicate there are no side-effects on reads
2:1	0h RO	**TYP:** Set to '00' to allow mapping anywhere in 32-bit address space

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**Memory Space Indicator (MEMSPACE):** Set to 0 for Memory Space

9.6 Sub System Identifiers (ESPI_SS) — Offset 2Ch

This register is initialized to logic 0 by the assertion of PLTRST#. This register can be written only once after PLTRST# de-assertion.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**Subsystem ID (SSID):** This is written by BIOS. No hardware action taken on this value.
15:0	0h RW/O	**Subsystem Vendor ID (SSVID):** This is written by BIOS. No hardware action taken on this value.

9.7 Capability List Pointer (ESPI_CAPP) — Offset 34h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 34h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD):** Reserved
7:0	0h RO	**Capability Pointer (CP):** Indicates the offset of the first Capability Item.

9.8 Capabilities List Pointer (CSXE_ESPI_CAPP) — Offset 40h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 40h	44 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
7:0	44h RO	**Capabilities Pointer (CAPP):** Indicates the pointer for the first entry in the capabilities list

9.9 MSI Message Control, Next Pointer and Capability ID (CSXE_ESPI_MSIMC_MSINP_MSICID) — Offset 44h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 44h	5005 h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD):** Reserved
24	0h RO	**Per Vector Masking Capable (PVMC):** This function does not support MSI per vector masking
23	0h RO	**64 bit address capable (XAC):** This function is not capable of sending 64 bit message address
22:20	0h RO	**Multiple Message Enable (MMEN):** Encoded number of interrupt vectors allocated by SW. Value of zero indicates one vector.
19:17	0h RO	**Multiple Message Capable (MMC):** Encoded number of interrupt vectors supported. Value of zero indicates one vector.
16	0h RW	**MSI Enable (MSIE):** If set (1) CXME MSI interrupt delivery is enabled. When this bit is cleared, prior to returning the configuration write completion, the device must send any pending MSI(s).
15:8	50h RO	**Next Item Pointer (NXTP):** Indicates the pointer for the next entry in the capabilities list. Hardwired to 50h to point to the PM Capability list.
7:0	5h RO	**Capability ID (CAPID):** Hardwired to 05h to indicate the linked list item as being the MSI Capability registers

9.10 MSI Message Address (CSXE_ESPI_MSIMA) — Offset 48h

MSI format is DW memory write in CSME root space with 32-bit addressing, 16 bit of data (upper word of data is zeros). The FW will only know the source (based on the FW programmed data field or vector) and it will need to read the cause/status register in device to get reason for MSI. FW clears after reading it.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 48h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**Message Address (MADDR):** DW aligned MSI message address.

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RO	**Reserved (RSVD):** Reserved

9.11 MSI Message Data (CSXE_ESPI_MSIMD) — Offset 4Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 4Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:0	0h RW	**Message Data (MDAT):** MSI Message Data

9.12 PCI Power Management Capability (CSXE_ESPI_PMCAP_PMNP_PMCID) — Offset 50h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 50h	40030001 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	8h RO	**PME Support (PMES):** This 5-bit field indicates the power states in which the function may assert PME#. A value of 0b for any bit indicates that the function is not capable of asserting the PME# signal while in that power state. bit(11) X XXX1b - PME# can be asserted from D0 bit(12) X XX1Xb - PME# can be asserted from D1 bit(13) X X1XXb - PME# can be asserted from D2 bit(14) X 1XXXb - PME# can be asserted from D3hot bit(15) 1 XXXXb - PME# can be asserted from D3cold
26	0h RO	**D2 Support (D2S):** This device does not support D2
25	0h RO	**D1 Support (D1S):** This device does not support D1
24:22	0h RO	**Aux Current (AUXC):** Not Applicable
21	0h RO	**Device Specific Initialization (DSI):** Indicates whether special initialization of this function is required (beyond the standard PCI configuration header) before the generic class device driver is able to use it. A 1 indicates that the function requires a device specific initialization sequence following transition to the D0 uninitialized state. Hardwired to 0 to indicate NO Device Specific Initialization is required.
20	0h RO	**Reserved (RSVD):** Reserved
19	0h RO	**PME Clock (PMECLK):** Not Applicable

Bit Range	Default & Access	Field Name (ID): Description
18:16	3h RO	**VER:** Value of 011b indicates that this function complies with rev 1.2 of PCI Power Management Interface Spec
15:8	0h RO	**Next Item Pointer (NXTP):** Indicates the pointer for the next entry in the capabilities list. Hardwired to 0 to indicate no more linked list item.
7:0	1h RO	**Capability ID (CAPP):** Hardwired to 01h to indicate the linked list item as being the PCI Power Management registers

9.13 PCI Power Management Control and Status (CSXE_ESPI_PMD_PMCSRBSE_PMCSR) — Offset 54h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 54h	8 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**DATA:** Not implemented. Hardwired to 0.
23:16	0h RO	**Reserved (RSVD):** Reserved
15	0h RW/1C/V/P	**PME Status (PMESTS):** This bit is set when the function would normally assert the PME signal independent of the state of the PME_En bit. Writing a 1 to this bit will clear it and cause the function to stop asserting a PME (if enabled). Writing a 0 has no effect. If the function supports PME from D3cold, then this bit is sticky and must be explicitly cleared by the operating system each time the operating system is initially loaded
14:13	0h RO	**Data Scale (DS):** Not Applicable
12:9	0h RO	**Data Select (DSEL):** Not Applicable
8	0h RW	**PME Enable (PMEEN):** A 1 enables the function to assert PME. When 0, PME assertion is disabled. This bit defaults to 0 if the function does not support PME generation from D3cold. If the function supports PME from D3cold then this bit is sticky and must be explicitly cleared by the operating system each time it is initially loaded.
7:4	0h RO	**Reserved (RSVD1):** Reserved
3	1h RO	**No Soft Reset (NSR):** When set to 1, this bit indicates that devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset. Configuration Context is preserved. Upon transition from the D3hot to the D0 Initialized state, no additional operating system intervention is required to preserve Configuration Context beyond writing the PowerState bits.
2	0h RO	**Reserved (RSVD2):** Reserved
1:0	0h RW	**Power State (PWRST):** This 2-bit field is used both to determine the current power state of a function and to set the function into a new power state. The definition of the field values is given below. 00b - D0 01b - D1 10b - D2 11b - D3hot If software attempts to write an unsupported, optional state to this field, the write operation must complete normally on the bus; however, the data is discarded and no state change occurs.

9.14 Peripheral Channel Serial IRQ Control (ESPI_PCSERIRQC) — Offset 64h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 64h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD):** Reserved

9.15 I/O Decode Ranges and I/O Enables (ESPI_IOD_IOE) — Offset 80h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 80h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_5):** Reserved
29	0h RW	**Microcontroller Enable #2 (ME2):** Enables decoding of I/O locations 4Eh and 4Fh to LPC.
28	0h RW	**SuperI/O Enable (SE):** Enables decoding of I/O locations 2Eh and 2Fh to LPC.
27	0h RW	**Microcontroller Enable #1 (ME1):** Enables decoding of I/O locations 62h and 66h to LPC.
26	0h RW	**KE:** Enables decoding of the keyboard I/O locations 60h and 64h to LPC.
25	0h RW	**High Gameport Enable (HGE):** Enables decoding of the I/O locations 208h to 20Fh to LPC.
24	0h RW	**Low Gameport Enable (LGE):** Enables decoding of the I/O locations 200h to 207h to LPC.
23:20	0h RO	**Reserved (RSVD_4):** Reserved
19	0h RW	**Floppy Drive Enable (FDE):** Enables decoding of the FDD range to LPC. Range is selected by LIOD.FDE
18	0h RW	**Parallel Port Enable (PPE):** Enables decoding of the LPT range to LPC. Range is selected by LIOD.LPT.
17	0h RW	**Com Port B Enable (CBE):** Enables decoding of the COMB range to LPC. Range is selected LIOD.CB.
16	0h RW	**Com Port A Enable (CAE):** Enables decoding of the COMA range to LPC. Range is selected LIOD.CA.
15:13	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**FDD:** The following table describes which range to decode for the FDD Port Bits Decode Range 0 3F0h - 3F5h, 3F7h (Primary) 1 370h - 375h, 377h (Secondary)
11:10	0h RO	**Reserved (RSVD_2):** Reserved
9:8	0h RW	**LPT:** The following table describes which range to decode for the LPT Port: Bits Decode Range 00 378h - 37Fh and 778h - 77Fh 01 278h - 27Fh (port 279h is read only) and 678h - 67Fh 10 3BCh - 3BEh and 7BCh - 7BEh 11 Reserved
7	0h RO	**Reserved (RSVD_1):** Reserved
6:4	0h RW	**ComB Range (CB):** The following table describes which range to decode for the COMB Port Bits Decode Range 000 3F8h - 3FFh (COM 1) 001 2F8h - 2FFh (COM 2) 010 220h - 227h 011 228h - 22Fh 100 238h - 23Fh 101 2E8h - 2EFh (COM 4) 110 338h - 33Fh 111 3E8h - 3EFh (COM 3)
3	0h RO	**Reserved (RSVD):** Reserved
2:0	0h RW	**ComA Range (CA):** The following table describes which range to decode for the COMA Port Bits Decode Range 000 3F8h - 3FFh (COM 1) 001 2F8h - 2FFh (COM 2) 010 220h - 227h 011 228h - 22Fh 100 238h - 23Fh 101 2E8h - 2EFh (COM 4) 110 338h - 33Fh 111 3E8h - 3EFh (COM 3)

9.16 LPC Generic I/O Range #1 (ESPI_LGIR1) — Offset 84h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 84h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23:18	0h RW	**Address[7:2] Mask (ADDR_MASK):** A 1 in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match. The corresponding bit in the Address field, below, is ignored. The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size.
17:16	0h RO	**Reserved (RSVD_1):** Reserved
15:2	0h RW	**ADDR:** DWord-aligned address. Note that PCH does not provide decode down to the word or byte level.
1	0h RO	**Reserved (RSVD_2):** Reserved
0	0h RW	**LPC Decode Enable (LDE):** When this bit is set to 1, then the range specified in this register is enabled for decoding to LPC.

9.17 LPC Generic I/O Range #2 (ESPI_LGIR2) — Offset 88h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 88h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23:18	0h RW	**Address[7:2] Mask (ADDR_MASK):** A 1 in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match. The corresponding bit in the Address field, below, is ignored. The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size.
17:16	0h RO	**Reserved (RSVD_1):** Reserved
15:2	0h RW	**ADDR:** DWord-aligned address. Note that PCH does not provide decode down to the word or byte level.
1	0h RO	**Reserved (RSVD_2):** Reserved
0	0h RW	**LPC Decode Enable (LDE):** When this bit is set to 1, then the range specified in this register is enabled for decoding to LPC.

9.18 LPC Generic I/O Range #3 (ESPI_LGIR3) — Offset 8Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 8Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23:18	0h RW	**Address[7:2] Mask (ADDR_MASK):** A 1 in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match. The corresponding bit in the Address field, below, is ignored. The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size.
17:16	0h RO	**Reserved (RSVD_1):** Reserved
15:2	0h RW	**ADDR:** DWord-aligned address. Note that PCH does not provide decode down to the word or byte level.
1	0h RO	**Reserved (RSVD_2):** Reserved
0	0h RW	**LPC Decode Enable (LDE):** When this bit is set to 1, then the range specified in this register is enabled for decoding to LPC.

9.19 LPC Generic I/O Range #4 (ESPI_LGIR4) — Offset 90h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 90h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23:18	0h RW	**Address[7:2] Mask (ADDR_MASK):** A 1 in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match. The corresponding bit in the Address field, below, is ignored. The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size.
17:16	0h RO	**Reserved (RSVD_1):** Reserved
15:2	0h RW	**ADDR:** DWord-aligned address. Note that PCH does not provide decode down to the word or byte level.
1	0h RO	**Reserved (RSVD_2):** Reserved
0	0h RW	**LPC Decode Enable (LDE):** When this bit is set to 1, then the range specified in this register is enabled for decoding to LPC.

9.20 USB Legacy Keyboard/Mouse Control (ESPI_ULKMC) — Offset 94h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 94h	2010 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15	0h RW/1C/V	**SMI Caused by End of Pass-through (SMIBYENDPS):** Indicates if the event occurred. Note that even if the corresponding enable bit is not set in the Bit 7, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Writing a 1 to this bit will clear the latch.
14	0h RO	**Reserved (RSVD_1):** Reserved
13	1h RW	**Reserved (RSVD_2):** Reserved
12	0h RO	**Reserved (RSVD_3):** Reserved
11	0h RW/1C/V	**SMI Caused by Port 64 Write (TRAPBY64W):** Indicates if the event occurred. Note that even if the corresponding enable bit is not set in the Bit 3, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Writing a 1 to this bit will clear the latch. Note that the A20Gate Pass-Through Logic allows specific port 64h Writes to complete without setting this bit.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW/1C/V	**SMI Caused by Port 64 Read (TRAPBY64R):** Indicates if the event occurred. Note that even if the corresponding enable bit is not set in the Bit 2, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Writing a 1 to this bit will clear the latch.
9	0h RW/1C/V	**SMI Caused by Port 60 Write (TRAPBY60W):** Indicates if the event occurred. Note that even if the corresponding enable bit is not set in the Bit 1, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Writing a 1 to this bit will clear the latch. Note that the A20Gate Pass-Through Logic allows specific port 60h Writes to complete without setting this bit.
8	0h RW/1C/V	**SMI Caused by Port 60 Read (TRAPBY60R):** Indicates if the event occurred. Note that even if the corresponding enable bit is not set in the Bit 0, then this bit will still be active. It is up to the SMM code to use the enable bit to determine the exact cause of the SMI#. Writing a 1 to this bit will clear the latch.
7	0h RW	**SMI at End of Pass-through Enable (SMIATENDPS):** May need to cause SMI at the end of a pass-through. Can occur if an SMI is generated in the middle of a pass through, and needs to be serviced later.
6	0h RO/V	**Pass Through State (PSTATE):** This read-only bit indicates that the state machine is in the middle of an A20GATE pass-through sequence. If software needs to reset this bit, it should set Bit 5 0.
5	0h RW	**A20Gate Pass-Through Enable (A20PASSEN):** When enabled, allows A20GATE sequence Pass-Through function. When enabled, a specific cycle sequence involving writes to port 60h and port 64h does not result in the setting of the SMI status bits.SMI# will not be generated, even if the various enable bits are set.
4	1h RW	**Reserved (RSVD_4):** Reserved
3	0h RW	**SMI on Port 64 Writes Enable (S64WEN):** When set, a 1 in bit 11 will cause an SMI event.
2	0h RW	**SMI on Port 64 Reads Enable (S64REN):** When set, a 1 in bit 10 will cause an SMI event.
1	0h RW	**SMI on Port 60 Writes Enable (S60WEN):** When set, a 1 in bit 9 will cause an SMI event.
0	0h RW	**SMI on Port 60 Reads Enable (S60REN):** When set, a 1 in bit 8 will cause an SMI event.

9.21 LPC Generic Memory Range (ESPI_LGMR) — Offset 98h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + 98h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Memory Address[31:16] (MADDR):** This field specifies a 64KB memory block anywhere in the 4GB memory space that will be decoded to LPC as standard LPC Memory Cycle if enabled.
15:1	0h RO	**Reserved (RSVD):** Reserved
0	0h RW	**LPC Memory Range Decode Enable (LGMRD_EN):** When this bit is set to 1, then the range specified in this register is enabled for decoding to LPC.

9.22 eSPI CS1# I/O Routing Enables (ESPI_CS1IORE) — Offset A0h

Note: This register is used to route fixed I/O transactions from the Host to the second Slave device (CS1#) over the Peripheral Channel on the eSPI bus. **Register Lock:** This register is locked in a single Slave configuration (soft-strap espi_cs1_en = 0b). All writes to this register must be dropped and reads to this register must return the default (reset) values. **Implementation / Usage Note:** PCIODE enables a given range if that is zero, all accesses to that range are U/R'd, irrespective of the settings for that range in this register. So PCIODE should be set/checked first. If a given range is enabled in PCIODE, then this register (PCCS1IORE) is used to select which eSPI Slave device to route the transactions to. When a second eSPI Slave device is not present, then PCCS1IORE is ignored and all IO transactions targeted to any range enabled in PCIODE are routed to CS0#.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + A0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD_1):** Reserved
14	0h RO	**Debug Port CS1# Routing Enable (DPRE):** Enables routing of I/O locations 80h, 84h-86h, 88h, 8Ch-8Eh, 90h, 94h-96h, 98h, 9Ch-9Eh to eSPI CS1#.
13	0h RO	**Microcontroller #2 CS1# Routing Enable (MRE2):** Enables routing of I/O locations 4Eh and 4Fh to eSPI CS1#.
12	0h RO	**SuperI/O CS1# Routing Enable (SRE):** Enables routing of I/O locations 2Eh and 2Fh to eSPI CS1#.
11	0h RO	**Microcontroller #1 CS1# Routing Enable (MRE1):** Enables routing of I/O locations 62h and 66h to eSPI CS1#.
10	0h RO	**Keyboard CS1# Routing Enable (KRE):** Enables routing of the keyboard I/O locations 60h and 64h to eSPI CS1#.
9	0h RO	**High Gameport CS1# Routing Enable (HGRE):** Enables routing of the I/O locations 208h to 20Fh to eSPI CS1#.
8	0h RO	**Low Gameport CS1# Routing Enable (LGRE):** Enables routing of the I/O locations 200h to 207h to eSPI CS1#.
7:4	0h RO	**Reserved (RSVD_2):** Reserved
3	0h RO	**Floppy Drive CS1# Routing Enable (FDRE):** Enables routing of the FDD range to eSPI CS1#. Range is selected by LIOD.FDE
2	0h RO	**Parallel Port CS1# Routing Enable (PPRE):** Enables routing of the LPT range to eSPI CS1#. Range is selected by LIOD.LPT.
1	0h RO	**Com Port B CS1# Routing Enable (CBRE):** Enables routing of the COMB range to eSPI CS1#. Range is selected by LIOD.CB.
0	0h RO	**Com Port A CS1# Routing Enable (CARE):** Enables routing of the COMA range to eSPI CS1#. Range is selected by LIOD.CA.

9.23 eSPI CS1# Generic I/O Range #1 (ESPI_CS1GIR1) — Offset A4h

This register has the same bit definitions as PCLGIR1. **Note:** This register is used to route variable IO transactions from the Host to the second Slave device (CS1#) over the Peripheral Channel on the eSPI bus. **Implementation Note:** If this range is enabled (LPC Decode Enable = 1b) in a single Slave configuration (espi_cs1_en soft-strap = 0b), any IO accesses to the range must by U/Rd by eSPI-MC.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + A4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23:18	0h RW	**Address[7:2] Mask (ADDR_MASK):** A 1 in any bit position indicates that any value in the corresponding address bit in a received cycle will be treated as a match. The corresponding bit in the Address field, below, is ignored. The mask is only provided for the lower 6 bits of the DWord address, allowing for decoding blocks up to 256 bytes in size.
17:16	0h RO	**Reserved (RSVD_1):** Reserved
15:2	0h RW	**ADDR:** DWord-aligned address. Note that PCH does not provide decode down to the word or byte level.
1	0h RO	**Reserved (RSVD_2):** Reserved
0	0h RW	**LPC Decode Enable (LDE):** When this bit is set to 1, then the range specified in this register is enabled for decoding to LPC.

9.24 eSPI CS1# Generic Memory Range #1 (ESPI_CS1GMR1) — Offset A8h

This register has the same bit definitions as PCLGMR. **Note:** This register is used to route memory transactions from the Host to the second Slave device (CS1#) over the Peripheral Channel on the eSPI bus. **Implementation Note:** If this range is enabled (LPC Decode Enable = 1b) in a single Slave configuration (espi_cs1_en soft-strap = 0b), any IO accesses to the range must by U/Rd by eSPI-MC.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + A8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Memory Address[31:16] (MADDR):** This field specifies a 64KB memory block anywhere in the 4GB memory space that will be decoded to LPC as standard LPC Memory Cycle if enabled.

Bit Range	Default & Access	Field Name (ID): Description
15:1	0h RO	**Reserved (RSVD):** Reserved
0	0h RW	**LPC Memory Range Decode Enable (LGMRD_EN):** When this bit is set to 1, then the range specified in this register is enabled for decoding to LPC.

9.25 FWH ID Select #1 (ESPI_PCFS1) — Offset D0h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + D0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD):** Reserved

9.26 FWH ID Select #2 (ESPI_PCFS2) — Offset D4h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + D4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD):** Reserved

9.27 BIOS Decode Enable (ESPI_BDE) — Offset D8h

Note that this register effects the BIOS decode regardless of whether the BIOS is resident on LPC or SPI. The concept of Feature Space does not apply to SPI-based flash. PCH simply decodes these ranges as memory accesses when enabled for the SPI flash interface.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + D8h	FFCF h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
15	1h RO	**F8-FF Enable (EF8):** Enables decoding of 512K of the following BIOS range: Data space: FFF80000h - FFFFFFFFh Feature space: FFB80000h - FFBFFFFFh
14	1h RW	**F0-F8 Enable (EF0):** Enables decoding of 512K of the following BIOS range: Data space: FFF00000h - FFF7FFFFh Feature space: FFB00000h - FFB7FFFFh
13	1h RW	**E8-EF Enable (EE8):** Enables decoding of 512K of the following BIOS range: Data space: FFE80000h - FFEFFFFFh Feature space: FFA80000h - FFAFFFFFh
12	1h RW	**E0-E8 Enable (EE0):** Enables decoding of 512K of the following BIOS range: Data space: FFE00000h - FFE7FFFFh Feature Space: FFA00000h - FFA7FFFFh
11	1h RW	**D8-DF Enable (ED8):** Enables decoding of 512K of the following BIOS range: Data space: FFD80000h - FFDFFFFFh Feature space: FF980000h - FF9FFFFFh
10	1h RW	**D0-D7 Enable (ED0):** Enables decoding of 512K of the following BIOS range: Data space: FFD00000h - FFD7FFFFh Feature space: FF900000h - FF97FFFFh
9	1h RW	**C8-CF Enable (EC8):** Enables decoding of 512K of the following BIOS range: Data space: FFC80000h - FFCFFFFFh Feature space: FF880000h - FF8FFFFFh
8	1h RW	**C0-C7 Enable (EC0):** Enables decoding of 512K of the following BIOS range: Data space: FFC00000h - FFC7FFFFh Feature space: FF800000h - FF87FFFFh
7	1h RW	**Legacy F Segment Enable (LFE):** This enables the decoding of the legacy 64KB range at F0000h - FFFFFh Note that decode for the BIOS legacy F segment is enabled by the LFE bit only, it is not affected by the GEN_PMCON_1.iA64_EN bit.
6	1h RW	**Legacy E Segment Enable (LEE):** This enables the decoding of the legacy 64KB range at E0000h - EFFFFh Note that decode for the BIOS legacy E segment is enabled by the LEE bit only, it is not affected by the GEN_PMCON_1.iA64_EN bit.
5:4	0h RO	**Reserved (RSVD):** Reserved
3	1h RW	**70-7F Enable (E70):** Enables decoding of 1MB of the following BIOS range: Data space: FF700000h - FF7FFFFFh Feature space: FF300000h - FF3FFFFFh
2	1h RW	**60-6F Enable (E60):** Enables decoding of 1MB of the following BIOS range: Data space: FF600000h - FF6FFFFFh Feature Space: FF200000h - FF2FFFFFh
1	1h RW	**50-5F Enable (E50):** Enables decoding of 1MB of the following BIOS range: Data space: FF500000h - FF5FFFFFh Feature space: FF100000h - FF1FFFFFh
0	1h RW	**40-4F Enable (E40):** Enables decoding of 1MB of the following BIOS range: Data space: FF400000h - FF4FFFFFh Feature space: FF000000h - FF0FFFFFh

9.28 BIOS Control (ESPI_BC) — Offset DCh

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + DCh	20 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_1):** Reserved
11	0h RW/L	**BIOS Write Reporting (Async-SMI) Enable (BWRE):** 1'b0: Disable reporting of BIOS Write event. 1'b1: Enable reporting of BIOS Write event (PCBC.BWRS = 1) using Async-SMI.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW/1C/V	**BIOS Write Status (BWRS):** HW sets this bit if a memory write access is detected to a protected BIOS range. 1'b0: Memory write to BIOS region not attempted or attempted with PCBC.WPD = 1. 1'b1: A memory write transaction to BIOS region has been received with PCBC.WPD = 0. Note: An Async-SMI message is generated to report this event if PCBC.BWRE is set. Note: SW must write a 1 to this bit to clear it, which will also deassert the Async-SMI, if PCBC.BWRE is set.
9	0h RO	**Reserved (RSVD_2):** Reserved
8	0h RW/1C/V	**BIOS Write Protect Disable Status (BWPDS):** HW sets this bit if configuration write access is detected to protected PCBC.WPD bit. 1'b0: No attempt has been made to set PCBC.WPD with PCBC.LE = 1. 1'b1: A configuration write request has been received to set PCBC.WPD (0 1) with PCBC.LE = 1. Note: An IOSF-SB Sync-SMI (Assert_SSMI) message is generated to report this event if HW sets this bit. The unsuccessful completion for the configuration write is returned upon receiving the SMI_Ack message response. Note: SW must write a 1 to this bit to clear it, which will also deassert the Sync-SMI (IOSF-SB Deassert_SSMI message is generated). Note: The Sync-SMI sets the PMC SMI_STS.TCO_STS register.
7	0h RW/L	**BIOS Interface Lock-Down (BILD):** When set, prevents BC.TS and BC.BBS from being changed. This bit can only be written from 0 to 1 once. BIOS Note: This bit is not backed up in the RTC well. This bit should also be set in the BUC register in the RTC device to record the last state of this value following a cold reset.
6	0h RW/V/L	**Boot BIOS Strap (BBS):** This field determines the destination of accesses to the BIOS memory range. Bits Description 0 SPI 1 LPC When SPI or LPC is selected, the range that is decoded is further qualified by other configuration bits. The value in this field can be overwritten by software as long as the BIOS Interface Lock-Down is not set.
5	1h RW/L	**Enable InSMM.STS (EISS):** When this bit is set, the BIOS region is not writable until SMM sets the InSMM.STS bit. Today BIOS Flash is writable if WPD is a 1. If this bit (5) is set, then WPD must be a 1 and InSMM.STS (0xFED3_0880(0)) must be 1 also. If this bit (5) is clear, then BIOS is writable based only on WPD = 1 and the InSMM.STS is a dont care.
4	0h RO/V	**Top Swap (TS):** When set, PCH will invert either A16, A17, or A18 for cycles going to the BIOS space (but not the Feature space) in the FWH. When cleared, PCH will not invert A16. If booting from LPC (FWH) or eSPI, then the Boot Block Size is fixed at 64KB and A16 is inverted if Top Swap is enabled. If booting from SPI, then the BOOT_BLOCK_SIZE soft strap determines if A16, A17, or A18 should be inverted if Top Swap is enabled. Note: If the Top-Swap strap is asserted, then this bit cannot be cleared by software. The strap jumper should be removed and the system rebooted. BIOS Note: This bit provides a read-only path to view the state of the Top Swap strap. It is backed up and driven from the RTC well. BIOS will need to program the corresponding register in the RTC well, which will be reflected in this register.
3	0h RO	**Reserved (RSVD):** Reserved
2	0h RO/V	**ESPI:** eSPI Enable Pin Strap (ESPI): This field determines the destination of accesses to the D31:F0 and related Fixed and Variable IO and Memory decode ranges, including BIOS memory range. 1'b0: LPC is the D31:F0 target. 1'b1: eSPI is the D31:F0 target. Note: This field, along with the PCBC.BBS strap setting, determines PCH configuration as specified in Table 11. Note: This field cannot be overwritten by software (unlike the PCBC.BBS field). Note: This bit is also reflected in the LPC (D31:F0) and SPI Flash (D31:F5) PCI Configuration register Offset DCh.
1	0h RW/L	**Lock Enable (LE):** When set, setting the WP bit will cause SMI. When cleared, setting the WP bit will not cause SMI. Once set, this bit can only be cleared by a PLTRST#. When this bit is set, EISS - bit (5) of this register is locked down.
0	0h RW	**Write Protect Disable (WPD):** When set, access to the BIOS space is enabled for both read and write cycles to BIOS. When cleared, only read cycles are permitted to the FWH or SPI flash. When this bit is written from a 0 to a 1 and the LE bit is also set, an SMI# is generated. This ensures that only SMM code can update BIOS.

9.29 Manufacturer's ID (ESPI_MANID) — Offset F8h

This register is assigned during the boot flow using the SetID message based on values found in the fuse block. All fields except MID will be reset by hardware to zero and set only by the SetID message. See Chapter 33 (DFT) for more information on the values that will be assigned in the fuse block for this component.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:31, F:0] + F8h	F00 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD_1):** Reserved
27:24	0h RO/V	**Dot portion of Process ID: (PD):** Indicates the dot process
23:16	0h RO/V	**Manufacturing Stepping Identifier (MSID):** This field is incremented for each stepping of the part. Note that this field can be used by software to differentiate steppings when the Stepping Revision ID may not change.
15:8	Fh RO/V	**Manufacturing Identifier (MID):** 0Fh = Intel
7:0	0h RO/V	**Process portion of process ID (PID):** Indicates the process. The dot portion of the process is reflected in bits (27:24)

§ §***<Enter text here>***

10 PMC Registers

This chapter documents the registers in Bus: 0, Device 13, Function 1.

10.1 DEVICEVENDORID - Device ID and Vendor ID Register (DEVVENDID) — Offset 0h

Device ID and Vendor ID provided by this register uniquely identifies the particular Device

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + 0h	A948086 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	A94h RO/V	**DEVICEVENDORID - Device ID Field (DEVICEID):** Device ID identifies the particular PCI device
15:0	8086h RO/V	**DEVICEVENDORID - Vendor ID Field (VENDORID):** Vendor ID is a unique ID provided by the PCI SIG which identifies the manufacturer of the device

10.2 STATUSCOMMAND — Offset 4h

Command register to programme interrupt disable bus master enable and Memory space enable. Status register to read the errors and aborts

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + 4h	100000 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved Field (RESERVED0):** Reserved
29	0h RW/1C/V	**STATUSCOMMAND- RMA Field (RMA):** Received Master Abort
28	0h RW/1C/V	**STATUSCOMMAND- RTA Field (RTA):** Received Target Abort
27:21	0h RO	**Reserved Field (RESERVED1):** Reserved
20	1h RO	**STATUSCOMMAND- Cap List Field (CAPLIST):** Capabilities List: Indicates that the controller contains a capabilities pointer list
19	0h RO	**STATUSCOMMAND- Intrerrupt Status Field (INTR_STATUS):** Interrupt Status: This bit reflects state of interrupt in the device

Bit Range	Default & Access	Field Name (ID): Description
18:16	0h RO	**Reserved Field (RESERVED2):** Reserved
15:11	0h RO	**Reserved Field (RESERVED3):** Reserved
10	0h RW	**STATUSCOMMAND- Interrupt Disable Field (INTR_DISABLE):** Interrupt Disable
9	0h RO	**Reserved Field (RESERVED4):** Reserved
8	0h RW	**STATUSCOMMAND- SERR Enable Field (SERR_ENABLE):** SERR Enable Not implemented
7:3	0h RO	**Reserved Field (RESERVED5):** Reserved
2	0h RW	**STATUSCOMMAND- BME Field (BME):** Bus Master Enable
1	0h RW	**STATUSCOMMAND- MSE Field (MSE):** Memory Space Enable
0	0h RW/V	**STATUSCOMMAND. IOSR FIELD (IOSE):** Controls a device's response to I/O Space accesses. A value of 0 disables the device response. A value of 1 allows the device to respond to I/O Space accesses. State after RST# is 0. NOTE: This bit does not exist in the PMC IOSF2OCP bridge. It is shadowed in the PSF3 fabric. Using /V in AccessType so PMC cluster validation does not assume this bit will read back what was written.

10.3 REVCLASSCODE — Offset 8h

Revision ID register identifies revision of particular device and Class Code register is used to identify generic function of the device

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + 8h	FF000000 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	FF0000h RO/V	**REVCLASSCODE - Class code Field (CLASS_CODES):** Class Code register is read-only and is used to identify the generic function of the device and in some cases a specific register-level programming interface
7:0	0h RO/V	**REVCLASSCODE - Revision ID Field (RID):** Revision ID identifies the revision of particular PCI device.

10.4 CLLATHEADERBIST — Offset Ch

Cache Line size as RW with def 0 Latency timer RW with def 0 Header type with Type 0 configuration header and Reserved BIST register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved Field (RESERVED0):** Reserved
23	0h RO	**CLLATHEADERBIST - MultiFunction Device Field (MULFNDEV):** Multi-Function Device
22:16	0h RO	**CLLATHEADERBIST - Header Type Field (HEADERTYPE):** Header Type: Implements Type 0 Configuration header
15:8	0h RO	**CLLATHEADERBIST - Latency Timer Field (LATTIMER):** Latency Timer:. This register is implemented as R/W with default as 0
7:0	0h RW	**CLLATHEADERBIST - Cache Line Size Field (CACHELINE_SIZE):** Cacheline Size

10.5 BAR — Offset 10h

Base Address Register low [31:2] type[2:1] in 32bit or 64bit addr range and memory space indicator [0]

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + 10h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**BAR -Base Address Field (BASEADDR):** Base Address Register Low Base address of the OCP fabric memory space. Taken from Strap values as ones
11:4	0h RO	**BAR -Size Field (SIZEINDICATOR):** Size Indicator RO Always returns 0 The size of this register depends on the size of the memory space
3	0h RO	**BAR -Prefetchable Field (PREFETCHABLE):** Prefetchable: Indicates that this BAR is not prefetchable
2:1	0h RO	**BAR -Type Field (TYPE):** If BAR_64b_EN is 0 then 00 indicates BAR lies in 32bit address range If BAR_64b_EN is 1 then 10 Indicates BAR lies in 64 bit address range
0	0h RO	**BAR - Message Space Field (MESSAGE_SPACE):** Memory Space Indicator: 0 indicates this BAR is present in the memory space.

10.6 BAR -Base Address Register High (BAR_HIGH) — Offset 14h

Base Address Register High enabled if [2:1] of BAR_type_LOW is 10

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + 14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**BAR -Base Address High Field (BASEADDR_HIGH):** Base Address high - MSB

10.7 BAR1 — Offset 18h

Base Address Register1 accesses to PCI configuration space and is always 4K type in [2:1] and memory space indicator in [0]

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + 18h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**BAR1 - Base Address Field (BASEADDR1):** Base Address1 This field is present if BAR1 is enabled through private configuration space.
11:4	0h RO	**BAR1 -Size Field (SIZEINDICATOR1):** Always is 0 as minimum size is 4K
3	0h RO	**BAR1 - Prefetchable Field (PREFETCHABLE1):** Prefetchable: Indicates that this BAR is not prefetchable.
2:1	0h RO	**BAR1 -Type Field (TYPE1):** If BAR_64b_EN is 0 then 00 indicates BAR lies in 32bit address range If BAR_64b_EN is 1 then 10 Indicates BAR lies in 64 bit address range
0	0h RO	**BAR1 - Message Space Field (MESSAGE_SPACE1):** Memory Space Indicator: 0 Indicates this BAR is present in the memory space

10.8 BAR1 -Base Address Register1 High (BAR1_HIGH) — Offset 1Ch

Base Address Register1 High enabled only if [2:1] of BAR1 register is 10

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + 1Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**BAR1 -Base Address High Field (BASEADDR1_HIGH):** Base Address: Base address of the OCP fabric memory space. Taken from Strap values as ones

10.9 BAR2 — Offset 20h

BAR -Base Address Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + 20h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**BASEADDR:**
1	0h RO	**RESERVED0:** Reserved
0	0h RO	**MESSAGE_SPACE:**

10.10 SUBSYSTEMID — Offset 2Ch

SVID register along with SID register is to distinguish subsystem from another

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**SUBSYSTEMID:** Subsystem ID: This register is implemented for any function that can be instantiated more than once in a given system.
15:0	0h RW/O	**SUBSYSTEMID -Subsystem Vendor Field (SUBSYSTEMVENDORID):** Subsystem Vendor ID: This register must be implemented for any function that can be instantiated more than once in a given system

10.11 EXPANSION ROM base address (EXPANSION_ROM_BASEADDR) — Offset 30h

EXPANSION ROM base address register is a RO indicates support for expansion ROMs

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + 30h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**EXPANSION ROM base address field (EXPANSION_ROM_BASE):** Expansion ROM Base Address: Value of all zeros indicates no support for Expansion ROM

10.12 CAPABILITYPTR — Offset 34h

Capabilities Pointer register indicates what the next capability is

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + 34h	80 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved Field (RESERVED0):** Reserved
7:0	80h RO	**CAPABILITYPTR - Capabilities Pointer Field (CAPPTR_POWER):** Capabilities Pointer: Indicates what the next capability is.

10.13 INTERRUPTREG — Offset 3Ch

Interrupt line Register isn't used in Bridge directly Interrupt Pin register reflects the IPIN value in private config space. Min_gnt register indicating the req of latency timers and max_lat register max latenc

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + 3Ch	100 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**INTERRUPTREG - Max Latency Field (MAX_LAT):** Value of 0 indicates device has no major requirements for the settings of latency timers
23:16	0h RO	**INTERRUPTREG - Min Gnt Field (MIN_GNT):** Value of 0 indicates device has no major requirements for the settings of latency timers // Re-Check Desc Padma
15:12	0h RO	**Reserved Field (RESERVED0):** Reserved
11:8	1h RO	**INTERRUPTREG - Interrupt Pin Field (INTPIN):** Interrupt Pin: Value in this register is reflected from the IPIN value in the private configuration space.
7:0	0h RW	**INTERRUPTREG - Int Line Field (INTLINE):** Interrupt Line: It is used to communicate to software the interrupt line to which the interrupt pin is connected

10.14 POWERCAPID — Offset 80h

PowerManagement Capability ID register points to next capability structure and power mgmnt capability with Power management capabilities register for PME support and version

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + 80h	48030001 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	9h RO	**POWERCAPID - PME Support Field (PMESUPPORT):** This 5-bit field indicates the power states in which the function can assert the PME#
26:19	0h RO	**Reserved Field (RESERVED0):** Reserved
18:16	3h RO	**POWERCAPID - Version Field (VERSION):** Indicates support for Revision 1.2 of the PCI Power Management Specification
15:8	0h RO	**POWERCAPID - Next Cap Field (NXTCAP):** Next Capability: Points to the next capability structure.
7:0	1h RO	**POWERCAPID - Power Capability ID Field (POWER_CAP):** Power Management Capability: Indicates this is power management capability

10.15 PMECTRLSTATUS — Offset 84h

power management control and status register to set and read PME status PME enable No Soft reset and power state

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + 84h	8 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved Field (RESERVED0):** Reserved
15	0h RW/1C/V	**PMECTRLSTATUS - PME Status Field (PMESTATUS):** PME Status
14:9	0h RO	**Reserved Field (RESERVED1):** Reserved
8	0h RW	**PMECTRLSTATUS - PME Enable Field (PMEENABLE):** PME Enable
7:4	0h RO	**Reserved Field (RESERVED2):** Reserved
3	1h RO	**PMECTRLSTATUS - No Soft Reset Field (NO_SOFT_RESET):** This bit indicates that devices transitioning from D3hot to D0 because of Powerstate commands do not perform an internal reset
2	0h RO	**Reserved Field (RESERVED3):** Reserved
1:0	0h RW	**PMECTRLSTATUS - Power State Field (POWERSTATE):** Power State: This field is used both to determine the current power state and to set a new power state

10.16 PCI DEVICE IDLE CAPABILITY RECORD (PCIDEVIDLE_CAP_RECORD) — Offset 90h

PCI Device Vendor Specific Capability register defines Vendor specific Capability ID, revision , length , next capability and CAPID

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + 90h	F0140009 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RO	**PCIDEVIDLE_CAP_REG - Vendor Cap Field (VEND_CAP):** Vendor Specific Capability ID
27:24	0h RO	**PCIDEVIDLE_CAP_REG - Revision IDField (REVID):** Revision ID of capability structure
23:16	14h RO	**PCIDEVIDLE_CAP_REG - Cap Length Field (CAP_LENGTH):** Vendor Specific Capability Length
15:8	0h RO	**PCIDEVIDLE_CAP_REG - Next Capability Field (NEXT_CAP):** Next Capability
7:0	9h RO	**PCIDEVIDLE_CAP_REG - Capability ID Field (CAPID):** Capability ID

10.17 DEVID VENDOR SPECIFIC REG (DEVID_VEND_SPECIFIC_REG) — Offset 94h

Extended Vendor capability register for VSEC Length revision and ID

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + 94h	1400010 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	14h RO	**DEVID VENDOR SPECIFIC REG - Vendor Specific ID Field (VSEC_LENGTH):** Vendor Specific Extended Capability Length
19:16	0h RO	**DEVID VENDOR SPECIFIC REG - Vendor Specific Revision Field (VSEC_REV):** Vendor specific Extended Capability revision
15:0	10h RO	**DEVID VENDOR SPECIFIC REG - Vendor Specific Length Field (VSECID):** Vendor Specific Extended Capability ID

10.18 D0I3_CONTROL_SW_LTR_MMIO_REG — Offset 98h

Software location pointer in MMIO space as an offset specified by BAR

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + 98h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**D0I3_CONTROL_SW_LTR_MMIO_REG - SW LTR Dword Offset Field (SW_LAT_DWORD_OFFSET):** SW LTR Update MMIO Offset Location (SWLTRLOC)
3:1	0h RO	**D0I3_CONTROL_SW_LTR_MMIO_REG - SW LTR BAR Number Field (SW_LAT_BAR_NUM):** Indicates that the SW LTR update MMIO location is always at BAR0
0	0h RO	**D0I3_CONTROL_SW_LTR_MMIO_REG - SW LTR Valid Field (SW_LAT_VALID):** This value is reflected from the SW LTR valid strap at the top level

10.19 DEVICE_IDLE_POINTER_REG — Offset 9Ch

Device IDLE pointer register giving details on Device MMIO offset location BAR NUM and D0i3 Valid Strap

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + 9Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**DEVICE_IDLE_POINTER_REG - D0i3 Dword Offset Field (DWORD_OFFSET):** contains the location pointer to the SW LTR register in MMIO space as an offset from the specified BAR
3:1	0h RO	**DEVICE_IDLE_POINTER_REG - BAR NUM Field (BAR_NUM):** Bar num: Indicates that the D0i3 MMIO location is always at BAR0
0	0h RO	**DEVICE_IDLE_POINTER_REG - D0i3 Valid Field (VALID):** Valid: This value is reflected from the D0i3 valid strap at the top level.

10.20 D0I3_MAX_POW_LAT_PG_CONFIG — Offset A0h

D0idle_Max_Power_On_Latency register set at boot and Power control enable register to enable communication with the PGCB block below the Bridge

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + A0h	800 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved Field (RESERVED0):** Reserved
21	0h RW	**D0I3_MAX_POW_LAT_PG_CONFIG - HAE Field (HAE):** Hardware Autonomous Enable
20	0h RO	**Reserved Field (RESERVED1):** Reserved
19	0h RW	**D0I3_MAX_POW_LAT_PG_CONFIG - Sleep Enable Field (SLEEP_EN):** Sleep Enable

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW	**D0I3_MAX_POW_LAT_PG_CONFIG - D3 Hen Field (PGE):** DEVIDLE Enable (DEVIDLEN): If ?1? then the function will power gate when idle and the DevIdle register (DevIdleC[2] = ?1?) is set.
17	0h RW	**D0I3_MAX_POW_LAT_PG_CONFIG - Device Idle En Field (I3_ENABLE):** D3-Hot Enable (D3HEN): If ?1?, then function will power gate when idle and the PMCSR[1:0] register in the function =?11? (D3).
16	0h RW	**D0I3_MAX_POW_LAT_PG_CONFIG - PMC Request Enable Field (PMCRE):** PMCRE: PMC Request Enable
15:13	0h RO	**Reserved Field (RESERVED2):** Reserved
12:10	2h RW/O	**D0I3_MAX_POW_LAT_PG_CONFIG - Power Latency Scale Field (POW_LAT_SCALE):** Power On Latency Scale
9:0	0h RW/O	**D0I3_MAX_POW_LAT_PG_CONFIG - Power Latency Value Field (POW_LAT_VALUE):** Power On Latency value

10.21 GEN_REGRW1 - General Purpose Read Write Register1 (GEN_REGRW1) — Offset B0h

General Purpose PCI Read Write Register1

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + B0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REGRW1 - General Purpose Read Write Field (GEN_REG_RW1):** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg1 Out of Band signal

10.22 GEN_REGRW2 — Offset B4h

General Purpose PCI Read Write Register2

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + B4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REGRW2 - General Purpose Read Write Field (GEN_REG_RW2):** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg2 Out of Band signal

10.23 GEN_REGRW3 — Offset B8h

General Purpose PCI Read Write Register3

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + B8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REGRW3 - General Purpose Read Write Field (GEN_REG_RW3):** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg3 Out of Band signal

10.24 GEN_REGRW4 — Offset BCh

General Purpose PCI Read Write Register4

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + BCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REGRW4 - General Purpose Read Write Field (GEN_REG_RW4):** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg4 Out of Band signal

10.25 GEN_INPUT_REG — Offset C0h

General Purpose Input Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + C0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**GEN_INPUT_REG - General Purpose Input Field (GEN_REG_INPUT_RW):** General Purpose Input Register: This register value reflects the value of oob_gen_input_pci Out of Band signal

10.26 Manufacturers ID (MANID) — Offset F8h

MAN ID Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:1] + F8h	4000F1C h

Bit Range	Default & Access	Field Name (ID): Description
31:0	4000F1Ch RO/V	**Manufacturers ID - MAN ID (MANID):** Manufacturer ID: Default value comes from straps.

10.27 THREAD_STATUS (P_CR_GT_THREAD_STATUS_0_2_0_GTTMMADR)—Offset 805Ch

Per thread status register THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 7FF07h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**THREAD_ACTIVE (THREAD_ACTIVE):** Tied to same value as CORE_STATUS.CORE_ACTIVE, updated by PMA.
30	0h RO/V	**VOTE_REQUEST (VOTE_REQUEST):** This bit will be set on TC0TC1 and cleared on TC1ETC7.
29:19	0h RO	**RESERVED_2 (RESERVED_2):** reserved
18:16	7h RO/V	**THREAD_WISH_ALLOW (THREAD_WISH_ALLOW):** This field contains the allowed core C-state limit. It is read by Ucode on C-state entry to clip its wish_state request. Pcode updates this field (via IO_CORE_DEMOTED_C1) based on demotion algorithm and: fuse, DFX, SW limits in PKG_C_STATE_LIMIT_REQ, probe mode and patch load (via Ucode Mbox).
15:12	Fh RW	**THREAD_WISH_SUB_STATE (THREAD_WISH_SUB_STATE):** Ucode updates this field with the desired Thread CState SubState.
11:8	Fh RW	**THREAD_WISH_STATE (THREAD_WISH_STATE):** Ucode updates this field with the parameters of the MWAIT instruction.
7	0h RO	**RESERVED_1 (RESERVED_1):** Reserved.
6:4	0h RW	**THREAD_TPD_STATE (THREAD_TPD_STATE):** Thread power down state
3	0h RO	**RESERVED_0 (RESERVED_0):** Reserved.
2:0	7h RW	**THREAD_STATE (THREAD_STATE):** Resolved thread_state updated by Ucode after it has completed entry/exit.

10.28 CORE_STATUS (P_CR_GT_CORE_STATUS_0_2_0_GTTMMADR)—Offset 8060h

Per core (including GT) status register.
Used by Ucode/GT and P-Unit/PMA HW to communicate core status to Pcode. THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 1100FF07h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**CORE_ACTIVE (CORE_ACTIVE):** Virtual signal indicating core is active. Value is updated by writes to RCSM_CORE_RESPONSE.CORE_ACTIVE from GT/GLM PMA. When set to '1', Pcode ignores value in CORE_STATE field and assumes C0, additionally P-Unit HW will delay fast path due to CORE_STATE field to non-C0 values until CORE_ACTIVE is '0'
30	0h RO/V	**WAKEUP_REQUEST (WAKEUP_REQUEST):** WAKEUP_REQUEST is set by interrupt wakes from T-Unit (via T2P_INT_WAKE) or Pcode via IO_PCODE_WAKEUP_REQUEST. WAKEUP_REQUEST is cleared by T-Unit (via T2P_INT_WAKE) when INT_READY vector is set. Fastpath (IO_FASTPATH_CORE_C) is set on WAKEUP_REQUEST, or when WAKEUP_REQUEST is '1' and CORE_IN_C3_C6 asserts.
29	0h RO	**RESERVED_3 (RESERVED_3):** Reserved
28	1h RO/V	**CORE_IN_C3_C6 (CORE_IN_C3_C6):** Virtual signal indicating core is in electrical (post RCSM flow) C3/C6. Value is updated by writes to RCSM_CORE_RESPONSE.CORE_IN_C3C6 from GT/GLM PMA. Must be asserted for WAKEUP_REQUEST to trigger fastpath.
27	0h WO	**PROBE_MODE_DONE (PROBE_MODE_DONE):** Sent by Ucode to inform P-Unit that probe mode seqeunce is complete, used to clear pending probe mode request in P-Unit HW.
26	0h RW	**DISABLE_WAKEUP_REQ (DISABLE_WAKEUP_REQ):** Reserved.
25	0h RO/V	**S1_ACK (S1_ACK):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
24	1h RW/V	**PM_BLOCK_REQ (PM_BLOCK_REQ):** Controls the status of the Interrupt Ready mask in the T-Unit. Set by Ucode before Piclet setup during C6 entry. On assertion of this bit P-Unit will send P2T_INT_CONTROL_ADDR.INT_READY_CLEAR to T-Unit. Any interrupts received after this point will cause T-Unit to send a interrupt wake request via T2P_INT_WAKE.SET_CORE_MASK Cleared by Ucode after Piclet replay during C6 exit. On deassertion of this bit, P-Unit will send P2T_INT_CONTROL_ADDR.INT_READY_SET to T-Unit. This will cause T-Unit to clear any pending interrupt wakes via T2P_INT_WAKE.CLEAR_CORE_MASK.
23	0h RO/V	**RFO_EN (RFO_EN):** For GT this bit indicates the status of read-for ownership (RFOs). If it is '1' RFOs are enabled and if it is '0' RFOs are disabled. This field only has meaning for the GT register instance. It is a don't care for the IA register instances.
22:16	0h RO	**RESERVED_2 (RESERVED_2):** Reserved
15:12	Fh RW	**CORE_WISH_SUB_STATE (CORE_WISH_SUB_STATE):** Ucode updates this field with the desired Core CState SubState.
11:8	Fh RW	**CORE_WISH_STATE (CORE_WISH_STATE):** Ucode updates this field with the parameters of the MWAIT instruction.
7	0h RO	**RESERVED_1 (RESERVED_1):** Reserved.
6:4	0h RW	**CORE_CPD_STATE (CORE_CPD_STATE):** Updated by Ucode on Core Power Down (CPD) exit or entry. CPD is used during GV Flows to quiesce core.
3	0h RO	**RESERVED_0 (RESERVED_0):** Reserved.
2:0	7h RW/V	**CORE_STATE (CORE_STATE):** Updated with the current core C-state, by Ucode on C6 entry/exit, at the end of the respective flow.

10.29 GT_SLICE_INFO (P_CR_GT_SLICE_INFO_0_2_0_GTTMMADR)—Offset 8064h

Status register describing GT slice state.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	Reserved.
10	0h RO/V	**UNSLICESTAT (UNSLICESTAT):** Status of GT unslice power plane • 0 = GT has unslice powered off • 1 = GT has unslice powered on
9:3	0h RO	Reserved.
2:0	0h RO/V	**SLICESTAT (SLICESTAT):** Status of GT power planes. • 000b = GT is powered off C6 or not yet booted • 001b = GT has slice 0 powered • 011b = GT has slices 0 and 1 powered • 111b = GT has slices 0, 1 and 2 powered

10.30 GT Hardware P-state Control Request (P_CR_GT_HWP_REQ_0_2_0_GTTMMADR)—Offset 8068h

This register is used as an interface for the graphics driver to communicate hardware managed P-state control requests and/or hints specific to the graphics domain. These requests are used by silicon power management firmware to implement SOC and platform level power and thermal control algorithms.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**Reserved3 (RSVD3):** This field is reserved for future use
23:16	0h RW	**Reserved2 (RSVD2):** This field is reserved for future use
15:8	0h RW	**Reserved1 (RSVD1):** This field is reserved for future use
7:0	0h RW	**Quality of Service P-state (QOS_RATIO):** GT Quality of Service P-state request. The GT driver programs this to define a desired performance floor. This floor is useful as a hint to power and thermal control algorithms to balance system resources appropriately in order to ensure a minimum graphics performance floor. Described in 16.67MHz reference clock units, e.g., a ratio of 24 results in a clock frequency of 400MHz.

10.31 GT_MEM_BOUND_COUNTER_0_2_0_GTTMMADR (P_CR_GT_MEM_BOUND_COUNTER_0_2_0_GTTMMADR)—Offset 8070h

GT memory bound counter. Holds the accumulated number of CS clks that GT has been waiting for memory grants.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**DATA (DATA):** Accumulated cycles GT has been waiting for memory grants.

10.32 GT_SQ_OCCUPANCY_0_2_0_GTTMMADR (P_CR_GT_SQ_OCCUPANCY_0_2_0_GTTMMADR)—Offset 8074h

Accumulated GT super queue (SQ) occupancy. This is accumulated by GT and pushed to P-Unit from the GT PMA.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**DATA (DATA):** Accumulated SQ occupancy.

10.33 GT_RW_DRAM_0_2_0_GTTMMADR (P_CR_GT_RW_DRAM_0_2_0_GTTMMADR)—Offset 8078h

This register contains the the sum of the cycles GT has read or written DRAM. It contains a 32-bit accumulation of data sent via the Pushbus. Values exceeding 32 bits will wrap around.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**DATA (DATA):** RW GT cycles to DRAM

10.34 GT_P_REQ (P_CR_GT_THREAD_P_REQ_0_2_0_GTTMMADR)—Offset 807Ch

GT Pstate request. This register is written by driver/GT with the desired P-State request. THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RW	**UNSLICE_RATIO (UNSLICE_RATIO):** UnSlice Ratio Multiple of 33.33MHz 2xclk 16 MHz 1xclk
22:14	0h RW	**SLICE_RATIO (SLICE_RATIO):** Slice Ratio Multiple of 33.33MHz 2xclk 16 MHz 1xclk
13:4	0h RO	**RESERVED_1 (RESERVED_1):** Reserved
3:0	0h RW	**SLICE_UNSLICE_POLICY (SLICE_UNSLICE_POLICY):** Specifies policy of slice and unslice ratios: • 0x0 = Use unslice ratio for slice also ignore slice ratio field • 0x1 = Maintain ratio between slice/unslice • 0x2 = Attempt to throttle slice only maintain unslice ratio

10.35 GT_ARAT_TTT (P_CR_GT_ARAT_TTT_0_2_0_GTTMMADR)—Offset 8080h

Always Running APIC Timer 'Timer Target Time' (TTT) value. This is an absolute desired wakeup time. GT copies the local TTT to the P-Unit on RC6 entry. Pcode will wake GT via IO_PCODE_WAKEUP_REQUEST, such that the wake is finished when the URT reaches the value in this register.
To account for wake delay, Pcode will also take into account the prewake value that can be written via driver into GTC6_PREWAKE_TIMER_0_2_0_GTTMMADR.
If ARAT is invalid, GT will write all 1's (infinity).
GT writes this register via the MMIO alias GT_ARAT_TTT_0_2_0_GTTMMADR THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Access Method

Type: MEM Register
(Size: 64 bits)

Device:
Function:

Default: 1FFFFFFFFFFFFFFFh

Bit Range	Default & Access	Field Name (ID): Description
63:61	0h RO	**RESERVED_0 (RESERVED_0):** Reserved
60:0	1FFFFFFFFFFFFFFFh RW	**DATA (DATA):** The 61 bits of the TTT. GT will update this field with TTT value before entering RC6.

10.36 GTC6_PREWAKE_TIMER_0_2_0_GTTMMADR (P_CR_GTC6_PREWAKE_TIMER_0_2_0_GTTMMADR)—Offset 8088h

This register is used in conjuction with GT_ARAT_TTT. It contains the enable and value for a prewake offset that the GT driver can program to prewake GT from C6 with reference to its TTT.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED_0 (RESERVED_0):** Reserved
15	0h RW	**TMR_ENABLE (TMR_ENABLE):** Enable for the GT prewake, the driver sets this bit to 1 to enable the offset.
14:0	0h RW	**TMR_VALUE (TMR_VALUE):** Prewake timer value in microseconds, this will be subtracted from the GT_ARAT_TTT.

10.37 GT_DISP_PWRON_0_2_0_GTTMMADR (P_CR_GT_DISP_PWRON_0_2_0_GTTMMADR)—Offset 8090h

Used by GT driver to control aspects of PHY power-on (MIPIO, eDP, DDI).

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**RESERVED_0 (RESERVED_0):** reserved
3	0h RW	**EDP_PWRREQ1P0_SUS (EDP_PWRREQ1P0_SUS):** GT Display EDP phy Power On. Written by GT driver to start EDP PHY initialization sequence.
2	0h RW	**MIPIO_RST_CTRL (MIPIO_RST_CTRL):** MIPIO reset control. Driver clears on reset exit which deasserts side_rst to MIPI PHY and starts periodic RCOMPS from P-Unit. Driver sets on PHY disabling, which should occur prior to dev2 D3.
1	0h RW	**CH1_PWRREQ1P0_SUS (CH1_PWRREQ1P0_SUS):** GT Display eDP Power On. Written by GT driver to start eDP PHY initialization sequence.
0	0h RW	**CH0_PWRREQ1P0_SUS (CH0_PWRREQ1P0_SUS):** GT Display DDI Power On. Written by GT driver to start DDI PHY initialization sequence.

10.38 GT_GFX_RC6_0_2_0_GTTMMADR (P_CR_GT_GFX_RC6_0_2_0_GTTMMADR)—Offset 8108h

Wrapping counter containing the total GT RC6 residency time since boot.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**RC6 (RC6):** GT RC6 residency, value is specified in 833.33ns increments. Counter will wrap around.

10.39 GTDRIVER_MAILBOX_INTERFACE_0_2_0_GTTMMADR (P_CR_GTDRIVER_MAILBOX_INTERFACE_0_2_0_GTTMMADR)—Offset 8124h

Control and Status register for the GFXDRIVERtoPCODE mailbox. This mailbox is implemented as a means for the GT Driver running on the IA cores to tune parameters for specific GFX workloads. This register is used in conjunction with GTDRIVER_MAILBOX_DATA_{HIGH and LOW}. THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S/V	**RUN_BUSY (RUN_BUSY):** SW may write to the two mailbox registers only when RUN_BUSY is clear(0). Setting RUN_BUSY to 1 will pend a Fast Path event to Pcode. After setting this bit SW will poll this bit until it is cleared. Alternatively PCODE can generate an interrupt to SW via GTDRIVER_P2G_EVENTS. PCODE will clear RUN_BUSY after updating the mailbox registers with the result and error code.
30:16	0h RO	Reserved.
15:8	0h RW/V	**PARAM1 (PARAM1):** This field is used to specify an additional parameter to extend the command when needed.
7:0	0h RW/V	**COMMAND (COMMAND):** This field contains the SW request command or the PCODE response code depending on the setting of RUN_BUSY.

10.40 GTDRIVER_MAILBOX_DATA_LOW_0_2_0_GTTMMADR (P_CR_GTDRIVER_MAILBOX_DATA_LOW_0_2_0_GTTMMADR)—Offset 8128h

Data register for lower 32b of data for the GTDRIVERtoPCODE mailbox. This mailbox is implemented as a means for the GTDriver running on the IA cores to tune parameters for specific GFX workloads. This register is used in conjunction with GTDRIVER_MAILBOX_INTERFACE. THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**DATA (DATA):** This field contains the lower 32b of data associated with specific commands.

10.41 GTDRIVER_MAILBOX_DATA_HIGH_0_2_0_GTTMMADR (P_CR_GTDRIVER_MAILBOX_DATA_HIGH_0_2_0_GTTMMADR)—Offset 812Ch

Data register for upper 32b of data for the GFXDRIVERtoPCODE mailbox. This mailbox is implemented as a means for the GT Driver running on the IA cores to tune parameters for specific GFX workloads. This register is used in conjunction with GTDRIVER_MAILBOX_INTERFACE. THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**DATA (DATA):** This field contains the upper 32b of data associated with specific commands.

10.42 P24C_PCODE_MAILBOX_INTERFACE_0_2_0_GTTMMADR (P_CR_P24C_PCODE_MAILBOX_INTERFACE_0_2_0_GTTMMADR)—Offset 8130h

Control and status register for the P24CtoPCODE mailbox. This mailbox is implemented as a way for the GT P24C to contact the P-unit and is accessible via MMIO. This register is used in conjunction with P24C_PCODE_MAILBOX_DATA. THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S/V	**RUN_BUSY (RUN_BUSY):** Software may write to the mailbox registers only when RUN_BUSY is clear(0). Setting RUN_BUSY to 1 will pend a fastpath even to Pcode. After setting this bit software polls until it is cleared. Pcode clears the bit after updating the mailbox registers with the result and error code.
30:24	0h RO	Reserved.
23:16	0h RW/V	**PARAM2 (PARAM2):** This field is used to specify a second additional parameter to extend the command when needed.
15:8	0h RW/V	**PARAM1 (PARAM1):** This field is used to specify an additional parameter to extend the command when needed.
7:0	0h RW/V	**COMMAND (COMMAND):** This field contains the SW request command or the PCODE response code depending on the setting of RUN_BUSY.

10.43 P24C_PCODE_MAILBOX_DATA_0_2_0_GTTMMADR (P_CR_P24C_PCODE_MAILBOX_DATA_0_2_0_GTTMMADR)—Offset 8134h

Data register for the P24CtoPCODE mailbox. This register is used in conjunction with P24C_PCODE_MAILBOX_INTERFACE THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**DATA (DATA):** This field contains the data associated with specific commands.

10.44 PCODE_P24C_MAILBOX_INTERFACE_0_2_0_GTTMMADR (P_CR_PCODE_P24C_MAILBOX_INTERFACE_0_2_0_GTTMMADR)—Offset 8138h

Control and status register for the PCODEtoP24C mailbox. This mailbox is implemented as a way for pcode to sendi commands to the P24C GT microcontroller. This register is used in conjunction with PCODE_P24C_MAILBOX_DATA. THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/0C/V	**RUN_BUSY (RUN_BUSY):** Pcode may write to the mailbox registers only when RUN_BUSY is clear(0). After setting RUN_BUSY to 1 PCode generates a PMLink command to shift in an uncore trap message to the GT P24C uC. GT P24C clears the RUN_BUSY bit after updating the mailbox registers with the result and error code. This clear of RUN_BUSY creates pends a fast path event to pcode.
30:24	0h RO	Reserved.
23:16	0h RW/V	**PARAM2 (PARAM2):** This field is used to specify a second additional parameter to extend the command when needed.
15:8	0h RW/V	**PARAM1 (PARAM1):** This field is used to specify an additional parameter to extend the command when needed.

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW/V	**COMMAND (COMMAND):** This field contains the PCODE request command or the GT P24C response code depending on the setting of RUN_BUSY. Command Encodings: 01h Cstate GoRequest for Duty Cycle Throttling Error Code Encodings

10.45 PCODE_P24C_MAILBOX_DATA_0_2_0_GTTMMADR (P_CR_PCODE_P24C_MAILBOX_DATA_0_2_0_GTTMMADR)—Offset 813Ch

Data register for the PCODEtoP24C mailbox. This register is used in conjunction with PCODE_P24C_MAILBOX_INTERFACE. This register is duplicated in the PCU IO space, XML changes must be made in both places

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**DATA (DATA):** This field contains the data associated with specific commands.

10.46 GT_PM_CONFIG_0_2_0_GTTMMADR (P_CR_GT_PM_CONFIG_0_2_0_GTTMMADR)—Offset 8140h

Interface for GT driver to configure power management.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	**RESERVED_0 (RESERVED_0):** Reserved
0	0h RW	**DOORBELL_EN (DOORBELL_EN):** Set by driver to enable doorbell mode. This must be set after GT is in and held in RC0 (force-wake). This field must be cleared prior to dev2 D3.

10.47 Graphics Interrupt Response Latency Tolerance (P_CR_GRAPHICS_INTERRUPT_RESPONSE_TIME_0_2_0_GTTMMADR)—Offset 8150h

This register is used by the graphics driver to communicate any interrupt latency tolerance (IRT/IRTL) requirements from the software. The P-unit is responsible for ensuring that the graphics worst-case interrupt response latency is always lower than what is programmed in this register.
Some components of worst-case exit latency for the graphics domain are outside the control of the P-unit, and therefore there is a physical floor below which the interrupt response tolerance may not be guaranteed. That floor is usually in the 100us range, but the number will vary by product and stepping.
IRT is used only for S0i3 entry if Dev2 is not in D3. If Dev2 is in D3, its assumed latency tolerance is infinity.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	Reserved.
15	0h RW	**VALID (VALID):** This field qualifies the validity of the Value field in this register. If the valid bit is zero, then it is assumed that software has no constraints on wake latency (i.e., it supports infinity).
14:13	0h RO	Reserved.
12:10	0h RW	**Multiplier (MULTIPLIER):** This field indicates the unit of measurement that is defined for the Value field in this register. The units are 2^(5*multiplier) • 000b = 1ns (2^0) • 001b = 32ns (2^5) • 010b = 1024ns (2^10) • 011b = 32.768us (2^15) • 100b = 1.048ms (2^20) • 101b = 33.55ms (2^25)
9:0	0h RW	**VALUE (VALUE):** This scalar is multiplied by the multiplier field to calculate the net latency tolerance. Ex., with a MULTIPLIER of 2 and a VALUE of 20, the net latency tolerance is 20 * (2^10) = 20480ns or 20.48us

10.48 GTDRIVER_P2G_EVENTS_0_2_0_GTTMMADR (P_CR_GTDRIVER_P2G_EVENTS_0_2_0_GTTMMADR)—Offset 8160h

This extended capability allows PCODE to send an interrupt notification upon completion of a mailbox command received from the GTDRIVER. It is enabled via the GTDriver Mailbox. PCODE will set the appropriate bit in this register to 1 and will then write to 0.2.0.GTTMMADR.PIM[PCU_MBOXE]. The GFX Driver will clear the appropriate bit in this register by writing a 1 to the bit. THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7	0h RW/1C/V	**EVENT7 (EVENT7):** Placeholder for Event
6	0h RW/1C/V	**EVENT6 (EVENT6):** Placeholder for Event
5	0h RW/1C/V	**EVENT5 (EVENT5):** Placeholder for Event
4	0h RW/1C/V	**EVENT4 (EVENT4):** Placeholder for Event
3	0h RW/1C/V	**EVENT3 (EVENT3):** Placeholder for Event
2	0h RW/1C/V	**EVENT2 (EVENT2):** Placeholder for Event
1	0h RW/1C/V	**EVENT1 (EVENT1):** Placeholder for Event
0	0h RW/1C/V	**EVENT0 (EVENT0):** This event indicates that the command previously sent by the GTDriver via the Mailbox mechanism is complete.

10.49 GTDRIVER_G2P_EVENTS_0_2_0_GTTMMADR (P_CR_GTDRIVER_G2P_EVENTS_0_2_0_GTTMMADR)—Offset 8164h

This extended capability allows the GTDriver to send a request to PCODE. The GTDriver will set the appropriate bit in this register to 1 when it wants to generate an event to PCODE. This will pend a Fast Path event to pcode. PCODE will clear the appropriate bit in this register after servicing the request. THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7	0h RW/1S/V	**EVENT7 (EVENT7):** Placeholder for Event
6	0h RW/1S/V	**EVENT6 (EVENT6):** Placeholder for Event
5	0h RW/1S/V	**EVENT5 (EVENT5):** Placeholder for Event
4	0h RW/1S/V	**EVENT4 (EVENT4):** Placeholder for Event
3	0h RW/1S/V	**EVENT3 (EVENT3):** Placeholder for Event
2	0h RW/1S/V	**EVENT2 (EVENT2):** Placeholder for Event
1	0h RW/1S/V	**EVENT1 (EVENT1):** Placeholder for Event
0	0h RW/1S/V	**EVENT0 (EVENT0):** Placeholder for Event

10.50 CORE_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_CORE_FREQUENCY_CAPABILITIES_0_2_0_GTTMMADR)—Offset 816Ch

PUNIT_MMIO: Core Frequency Capabilities
This register describes the frequency capabilities of the IA cores. Units are 100MHz multiplied by the ratio.
Minimum and maximum ratio fields are initialized by pCode at reset. Last resolved ratio is updated upon changes to the processing system frequency. The efficient ratio is determined by firmware and may be updated dynamically depending on firmware support.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO/V	**LAST_RESOLVED_FREQ (LAST_RESOLVED_FREQ):** Last resolved ratio for the IA cores. Units are 100MHz multiplied by the ratio. This value is updated dynamically whenever the IA core frequency changes.

Bit Range	Default & Access	Field Name (ID): Description
23:16	0h RO/V	**MAX_SUPPORTED_FREQ (MAX_SUPPORTED_FREQ):** Maximum ratio for the IA cores. Units are 100MHz multiplied by the ratio.
15:8	0h RO/V	**EFFICIENT_FREQ (EFFICIENT_FREQ):** Firmware-calculated efficient ratio for the IA cores. Units are 100MHz multiplied by the ratio.
7:0	0h RO/V	**MIN_SUPPORTED_FREQ (MIN_SUPPORTED_FREQ):** Minimum supported ratio for the IA cores. Units are 100MHz multiplied by the ratio.

10.51 GRAPHICS_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR (P_CR_GRAPHICS_FREQUENCY_CAPABILITIES_0_2_0_GTTMMADR)—Offset 8170h

PUNIT_MMIO: Graphics Engine Frequency Capabilities
This register describes the frequency capabilities of the integrated graphics engine. Units are 16.67MHz multiplied by the ratio.
Minimum and maximum ratio fields are initialized by pCode at reset. Last resolved ratio is updated upon changes to the integrated graphics engine frequency. The efficient ratio is determined by firmware and may be updated dynamically depending on firmware support.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO/V	**LAST_RESOLVED_FREQ (LAST_RESOLVED_FREQ):** Last resolved ratio for the integrated graphics engine. Units are 16.67MHz multiplied by the ratio. This value is updated dynamically whenever the graphics engine frequency changes.
23:16	0h RO/V	**MAX_SUPPORTED_FREQ (MAX_SUPPORTED_FREQ):** Maximum supported ratio for the integrated graphics engine. Units are 16.67MHz multiplied by the ratio.
15:8	0h RO/V	**EFFICIENT_FREQ (EFFICIENT_FREQ):** Firmware-calculated efficient ratio for the integrated graphics engine. Units are 16.67MHz multiplied by the ratio.
7:0	0h RO/V	**MIN_SUPPORTED_FREQ (MIN_SUPPORTED_FREQ):** Minimum supported ratio for the integrated graphics engine. Units are 16.67MHz multiplied by the ratio.

10.52 SYSTEM_AGENT_FREQUENCY_CAPABILITIES_0_0_0_MCHBAR

(P_CR_SYSTEM_AGENT_FREQUENCY_CAPABILITIES_0_2_0_GTTMMADR)—Offset 8174h

PUNIT_MMIO: System Agent Frequency Capabilities
This register describes the frequency capabilities of the System Agent. Units are 16.666MHz multiplied by the ratio.
Last resolved ratio is updated upon changes to the System Agent frequency.

Access Method

Type: MEM Register (Size: 32 bits)	**Device:** **Function:**

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO/V	**LAST_RESOLVED_RATIO (LAST_RESOLVED_RATIO):** Last resolved System Agent ratio, in units of 16.666MHz.
23:16	0h RO/V	**MAX_RATIO (MAX_RATIO):** max ratio
15:8	0h RO/V	**RESERVED_1 (RESERVED_1):** Reserved
7:0	0h RO/V	**MIN_RATIO (MIN_RATIO):** min ratio

10.53 Memory Frequency Status (P_CR_FAR_MEMORY_FREQUENCY_CAPABILITIES_0_2_0_GTTMMADR)—Offset 8178h

This register reports out the LPDDR memory frequency. The actual capabilities of the SOC with respect to LPDDR frequency is described in the MEMSS_FREQUENCY_CAPABILITIES register.

Access Method

Type: MEM Register (Size: 32 bits)	**Device:** **Function:**

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO/V	**Last Resolved Memory Frequency (LAST_RESOLVED_RATIO):** This field reports out the LPDDR memory frequency in integer multiple of 133.33MHz. This register reflects what BIOS has programmed as the default LPDDR frequency in products that do not support run-time memory frequency control. For products supporting run-time memory frequency control, this field describes the last resolved frequency.
23:16	0h RO/V	**RESERVED_2 (RESERVED_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO/V	**RESERVED_1 (RESERVED_1):** Reserved
7:0	0h RO/V	**RESERVED_0 (RESERVED_0):** Reserved

10.54 GT_PERF_LIMIT_REASONS (P_CR_GT_PERF_LIMIT_REASONS_0_2_0_GTTMMADR)—Offset 8184h

This register reports reasons for performance limitations on the integrated graphics engine. Status bits are an instantaneous indication of an active constraint. Log bits indicate that a constraint was enforced since the log bit was last cleared.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/0C/V	**QOS_LOG (QOS_LOG):** Logged indication that frequency was clamped below the software-defined quality-of-service floor. This bit is set by firmware, and is clearable by software.
30	0h RW/0C/V	**MAX_EFFICIENCY_FREQ_LOG (MAX_EFFICIENCY_FREQ_LOG):** Logged indication that frequency was clamped below the firmware-calculated maximum efficiency frequency. This bit is set by firmware, and is clearable by software.
29	0h RW/0C/V	**SPARE13_LOG (SPARE13_LOG):** Spare log bit. This bit is set by firmware, and is clearable by software.
28	0h RW/0C/V	**EDP_LOG (EDP_LOG):** Logged indication that frequency was clamped due to the package-level Electrical Design Point constraint. This bit is set by firmware, and is clearable by software.
27	0h RW/0C/V	**SPARE11_LOG (SPARE11_LOG):** Spare log bit. This bit is set by firmware, and is clearable by software.
26	0h RW/0C/V	**VR_THERMALERT_LOG (VR_THERMALERT_LOG):** Logged indication that frequency was clamped due to a voltage regulator thermal excursion. This bit is set by firmware, and is clearable by software.
25	0h RW/0C/V	**SPARE9_LOG (SPARE9_LOG):** Spare log bit. This bit is set by firmware, and is clearable by software.
24	0h RW/0C/V	**SPARE8_LOG (SPARE8_LOG):** Spare log bit. This bit is set by firmware, and is clearable by software.
23	0h RW/0C/V	**SPARE7_LOG (SPARE7_LOG):** Spare log bit. This bit is set by firmware, and is clearable by software.

Bit Range	Default & Access	Field Name (ID): Description
22	0h RW/0C/V	**SPARE6_LOG (SPARE6_LOG):** Spare log bit. This bit is set by firmware, and is clearable by software.
21	0h RW/0C/V	**SPARE5_LOG (SPARE5_LOG):** Spare log bit. This bit is set by firmware, and is clearable by software.
20	0h RW/0C/V	**SPARE4_LOG (SPARE4_LOG):** Spare log bit. This bit is set by firmware, and is clearable by software.
19	0h RW/0C/V	**PL2_LOG (PL2_LOG):** Logged indication that frequency was clamped due to a package-level PL2 excursion. This bit is set by firmware, and is clearable by software.
18	0h RW/0C/V	**PL1_LOG (PL1_LOG):** Logged indication that frequency was clamped due to a package-level PL1 excursion. This bit is set by firmware, and is clearable by software.
17	0h RW/0C/V	**THERMAL_LOG (THERMAL_LOG):** Logged indication that frequency was clamped due to a thermal excursion. This bit is set by firmware, and is clearable by software.
16	0h RW/0C/V	**PROCHOT_LOG (PROCHOT_LOG):** Logged indication that frequency was clamped due to PROCHOT assertion. This bit is set by firmware, and is clearable by software.
15	0h RO/V	**QOS_STATUS (QOS_STATUS):** Frequency is limited below the operating system or driver Quality-of-Service floor.
14	0h RO/V	**MAX_EFFICIENCY_FREQ_STATUS (MAX_EFFICIENCY_FREQ_STATUS):** Frequency is limited below the maximum efficiency frequency.
13	0h RO/V	**SPARE13_STATUS (SPARE13_STATUS):** Frequency is limited due to ratio change transition attenuation (MCT, prevents frequent ratio changes due to core C-state entry/exit).
12	0h RO/V	**EDP_STATUS (EDP_STATUS):** Frequency is limited due to a package-level EDP constraint.
11	0h RO/V	**SPARE11_STATUS (SPARE11_STATUS):** Spare status bit.
10	0h RO/V	**VR_THERMALERT_STATUS (VR_THERMALERT_STATUS):** Frequency is limited due to a voltage regulator thermal excursion.
9	0h RO/V	**SPARE9_STATUS (SPARE9_STATUS):** Spare status bit.
8	0h RO/V	**SPARE8_STATUS (SPARE8_STATUS):** Spare status bit.
7	0h RO/V	**SPARE7_STATUS (SPARE7_STATUS):** Spare status bit.
6	0h RO/V	**SPARE6_STATUS (SPARE6_STATUS):** Spare status bit.
5	0h RO/V	**SPARE5_STATUS (SPARE5_STATUS):** Spare status bit.
4	0h RO/V	**SPARE4_STATUS (SPARE4_STATUS):** Spare status bit.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO/V	**PL2_STATUS (PL2_STATUS):** Frequency is limited due to a package-level PL2 excursion.
2	0h RO/V	**PL1_STATUS (PL1_STATUS):** Frequency is limited due to a package-level PL1 excursion.
1	0h RO/V	**THERMAL_STATUS (THERMAL_STATUS):** Frequency is limited due to thermal excursion.
0	0h RO/V	**PROCHOT_STATUS (PROCHOT_STATUS):** Frequency is limited due to external PROCHOT assertion.

10.55 GTDRIVER_HWP_REQUEST_0_2_0_GTTMMADR (P_CR_GTDRIVER_HWP_REQUEST_0_2_0_GTTMMADR)—Offset 8190h

PUNIT_MMIO: GTDRIVER_HWP_REQUEST
This register allows the graphics driver to dictate the minimum and maximum performance of the IA cores. The driver may also change the behavior of any autonomous IA P-state controller, if supported by firmware. Non-zero values override default firmware behavior.

Access Method

Type: MEM Register (Size: 64 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:42	0h RW	**RESERVED_0 (RESERVED_0):** Reserved
41:32	0h RW	**ACTIVITY_WINDOW (ACTIVITY_WINDOW):** This field defines the time window over which the autonomous IA P-state controller (if supported) will manage IA core frequency. Units are microseconds.
31:24	0h RW	**AGGRESSIVENESS (AGGRESSIVENESS):** This field defines the performance/energy bias for the autonomous IA P-state controller, if supported.
23:16	0h RW	**MULTIPLIER (MULTIPLIER):** This field defines the IA frequency target: IA ratio = MULTIPLER * GT ratio. The format is U8.3.5 (e.g. 1.0 is 8h'20). The IA ratio is described in 100MHz units and GT ratio is in 16.67MHz units. E.g., to achieve IA 1.6GHz (ratio 16) at GT 400MHz (ratio 24), the desired multiplier would be 0.67 (0x15).
15:8	0h RW	**MAX_PERF (MAX_PERF):** Maximum IA core frequency allowed by the graphics driver. The field is defined as a ratio, multiply by 100MHz to convert to IA core frequency.

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**MIN_PERF (MIN_PERF):** Minimum IA core frequency allowed by the graphics driver. The field is defined as a ratio, multiply by 100MHz to convert to IA core frequency.

10.56 GT_VIDEO_BUSYNESS_0_2_0_GTTMMADR (P_CR_GT_VIDEO_BUSYNESS_0_2_0_GTTMMADR)—Offset 819Ch

GT video busy counter. Holds the accumulated number of CS clks that GT video functions have been busy.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**DATA (DATA):** Accumulated cycles GT video functions have been busy.

RID - Bridge revision ID Register (RID)—Offset 30h

register contains the Bridge Revision ID

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved Field (Reserved0):** Reserved
7:0	0h RO	**RID - Bridge revision ID Field (RID):** Revision ID of the Bridge.

10.57 GEN_REGRW1 - General Purpose register (GEN_REGRW1)—Offset 600h

General Purpose PRV RW Register 1

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REGRW1 - General Purpose Field (GEN_REG_RW1):** This register value is brought out as oob_gen_prv_rw_reg1 out of band signal

10.58 GEN_REGRW2 - General Purpose register (GEN_REGRW2)—Offset 604h

General Purpose PRV RW Register 2

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REGRW2 - General Purpose Field (GEN_REG_RW2):** This register value is brought out as oob_gen_prv_rw_reg2 out of band signal

10.59 GEN_REGRW3 - General Purpose register (GEN_REGRW3)—Offset 608h

General Purpose PRV RW Register 3

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REGRW3- General Purpose Field (GEN_REG_RW3):** This register value is brought out as oob_gen_prv_rw_reg3 out of band signal

10.60 GEN_REGRW4 - General Purpose register (GEN_REGRW4)—Offset 60Ch

General Purpose PRV RW Register 4

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REGRW4 - General Purpose Field (GEN_REG_RW4):** This register value is brought out as oob_gen_prv_rw_reg4 out of band signal

Power Management 1 Status and Enable (PM1_STS_EN)—Offset 0h

Access Method

Type: IO Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (reserved4):** Reserved.
30	0h RW	**PCI Express Wake Disable (pciexp_wake_dis):** This bit is a scratchbit. The value written can be read back but there is not functionality behind the bit. Windows and Linux have confirmed that they dont use this bit but require it to save and return the written state. Historical reference only: This bit disables the inputs to the PCIEXP_WAKE_STS bit in the PM1 Status register from waking the system. Modification of this bit has no impact on the value of the PCIEXP_WAKE_STS bit
29	0h RO	**Reserved: USB Clockless Wake Enable (rsvd_usb_clkless_en):** Reference only: This bit enables the inputs to the USB_CLKLESS_STS bit in the PM1 Status register to wake the system. Modification of this bit has no impact on the value of the USB_CLKLESS_STS bit. On some prior platforms, this field is reset on RSMRST_B, a reset signal based on a RSMRST# pin that indicates the suspend/resume voltages are stable.
28:27	0h RO	**Reserved (reserved5):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW/V	**RTC Alarm Enable (rtc_en):** This is the RTC alarm enable bit. It works in conjunction with the SCI_EN bit: RTC_EN | SCI_EN | Effect when RTC_STS is set 0 | x | No SMI# or SCI. If system was in S1-S5, no wake even occurs. 1 | 0 | SMI#. If system was in S1-S5, then a wake event occurs before the SMI#. 1 | 1 | SCI. If system was in S1-S5, then a wake event occurs before the SCI. Note: To clarify, If rtc_en is set and and RTC_STS gets set during S0/S0ix and SCI/SMI will be sent depending on the state of SCI_EN and GBL_SMI_EN. In addition to its normal reset conditions, PMC also clears this bit due to emergency shutdown events: - Power button override - CPU thermal trip.
25	0h RO	**Reserved (reserved6):** Reserved.
24	0h RW/V	**Power Button Enable (pwrbtn_en):** This bit is the power button enable. It works in conjunction with these bits: PWRBTN_EN | SCI_EN | Effect when PWRBTN_STS is set 0 | x | No SMI#, SCI or S0ix wake. 1 | 0 | SMI# and an S0ix wake. 1 | 1 | SCI and an S0ix wake. Note: PWRBTN_EN has no effect on the PWRBTN_STS bit being set by the assertion of the power button. The Power Button is always enabled as an SX Wake event.
23:22	0h RO	**Reserved (reserved7):** Reserved.
21	0h RW	**Global Enable (gbl_en):** The global enable bit. When both the GBL_EN and the GBL_STS are set, PMC generates an SCI.
20:17	0h RO	**Reserved (reserved8):** Reserved.
16	0h RO	**Reserved: Timer Overflow Interrupt Enable (rsvd_tmrof_en):** Reference only: This is the timer overflow interrupt enable bit. It works in conjunction with the SCI_EN bit: TMROF_EN | SCI_EN | Effect when TMROF_STS is set. 0 | x | No SMI# or SCI 1 | 0 | SMI# 1 | 1 | SCI

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C/V	**Wake Status (wak_sts):** This bit is set when the system is in one of the Sleep states (via the SLP_EN bit) and an enabled Wake event occurs. Upon setting this bit, the PMC will transition the system to the ON state. This bit can only be set by hardware and can only be cleared by writing a one to this bit position. If a power failure occurs (such as removed batteries) without the SLP_EN bit set, the WAK_STS bit will not be set when the power returns if the AFTER_G3 bit is 0. If the AFTER_G3 bit is 1, then the WAK_STS bit will be set after waking from a power failure. If necessary, the BIOS can clear the WAK_STS bit in this case. According to ACPI spec 5.0 If the system does not support the S1 sleeping state, the WAK_STS bit can always return zero.
14	0h RO	**Reserved: PCI Express Wake Status (rsvd_pciexp_wake_sts):** Reference only: This bit is set by hardware to indicate that the system woke due to a PCI Express wakeup event. This event can be caused by the PCI Express WAKE# pins (PMU_WAKE_B, PCI_WAKE1_B, PCI_WAKE2_B, PCI_WAKE3_B) being active, or one or more of the PCI Express ports being in beacon state, or receipt of a PCI Express PME message at root port. This bit should only be set when one of these events causes the system to transition from a non-S0 system power state to the S0 system power state. This bit is set independent of the PCIEXP_WAKE_DIS bit. Software writes a 1 to clear this bit. If one of the WAKE# pins is still active during the write or one or more PCI Express ports is in the beacon state or PME message received indication is not cleared in the root port, then the bit will remain Power Management active (i.e. all inputs to this bit are level sensitive). Note: This bit does not itself cause a wake event or prevent entry to a sleeping state. Thus if the bit is 1 and the system is put into a sleeping state, the system will not automatically wake.
13	0h RO	**Reserved: USB clockless Wake Status (rsvd_usb_clkless_sts):** Reference only: This bit is set by hardware to indicate that the system woke due to change in USB serial lines. This bit is set independent of the USB_CLKLESS_EN bit. Software writes a 1 to clear this bit.
12	0h RO	**Reserved (rsvd):** Reserved.
11	0h RW/1C/V	**Power Button Override (pwrbtnor_sts):** This bit is set any time a Power Button Override Event occurs (i.e. the power button is pressed for at least 4 consecutive seconds- These events cause an unconditional transition to the S5 state. The BIOS or SCI handler clears this bit by writing a 1 to it. Note that this bit is preserved through power failures. If it is still asserted when the global SCI_EN is set to '1' then an SCI will be generated.
10	0h RW/1C/V	**RTC Status (rtc_sts):** This bit is set when the RTC generates an alarm (assertion of the IRQ8# signal), and is not affected by any other enable bit. See RTC_EN for the effect when RTC_STS goes active. This bit is only set by hardware and can only be reset by writing a one to this bit position.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RO	**Reserved (reserved1):** Reserved.
8	0h RW/1C/V	**Power Button Status (pwrbtn_sts):** This bit is set when the PMU_PWRBTN_B signal is asserted (low), independent of any other enable bit. See PWRBTN_EN for the effect when PWRBTN_STS goes active. PWRBTN_STS is always a wake event. This bit is only set by hardware and can be cleared by software writing a one to this bit position. If the PMU_PWRBTN_B signal is held low for more than 4 seconds, the PMC clears the PWRBTN_STS bit, sets the PWRBTNOR_STS bit, the system transitions to the S5 state, and only PMU_PWRBTN_B is enabled as a wake event. If PWRBTN_STS bit is cleared by software while the PMU_PWRBTN_B pin is still held low, this will not cause the PWRBTN_STS bit to be set. The PMU_PWRBTN_B signal must go inactive and active again to set the PWRBTN_STS bit. Note that the CPU Thermal Trip and the Internal Thermal Sensors' Catastrophic Condition result in behavior matching the Power Button Override, which includes clearing this bit.
7:6	0h RO	**Reserved (reserved2):** Reserved.
5	0h RW/1C/V	**Global Status (gbl_sts):** This bit is set when an SCI is generated due to the BIOS wanting the attention of theSCI handler. BIOS has a corresponding bit, BIOS_RLS, which will cause an SCI and set this bit. The SCI handler should then clear this bit by writing a 1 to it. This bit will not cause wake events or SMI#. This bit is not effected by SCI_EN. Note: GBL_STS being set will cause an SCI, even if the SCI_EN bit is not set. Software must take great care not to set the BIOS_RLS bit (which causes GBL_STS to be set) if the SCI handler is not in place.
4:1	0h RO	**Reserved (rserved3):** Reserved.
0	0h RO	**Reserved: Timer Overflow Status (rsvd_tmrof_sts):** Reference only: This is the timer overflow status bit. This bit gets set anytime bit 22 of the 24 bit timer goes low (bits are counted from 0 to 23). This will occur every 2.3435 seconds. See TMROF_EN for the effect when TMROF_STS goes active. Software clears this bit by writing a 1 to it.

10.61 Power Management 1 Control (PM1_CNT)—Offset 4h

Access Method

Type: IO Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**Reserved (rsvd):** Reserved.
13	0h WO	**Sleep Enable (slp_en):** This is a write-only bit and reads to it always return a zero. Setting this bit causes the system to sequence into the Sleep state defined by the SLP_TYP field.
12:10	0h RW/V	**Sleep Type (slp_typ):** This 3-bit field defines the type of Sleep the system should enter when the SLP_EN bit is set to 1. Bits | Mode | Typical Mapping 000 | ON | S0 001 | Reserved 010 | Reserved 011 | Reserved 100 | Reserved 101 | Suspend-To-RAM | S3 110 | Suspend-To-Disk | S4 111 | Soft Off | S5
9:3	0h RO	**Reserved (reserved1):** Reserved.
2	0h WO	**Global RLS (gbl_rls):** This bit is used by the ACPI software to raise an event to the BIOS software. BIOS software has a corresponding enable and status bits to control its ability to receive ACPI events. This bit always reads as 0.
1	0h RW	**BM RLD (bm_rld):** This bit is treated as a scratch pad bit.
0	0h RW	**SCI Enable (sci_en):** Selects the SCI interrupt or the SMI# for various events. When this bit is 1, then the events will generate an SCI interrupt. When this bit is 0, these events will generate an SMI#.

10.62 Power Management 1 Timer (PM1_TMR)—Offset 8h

Access Method

Type: IO Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (rsvd):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
23:0	0h RO	**Reserved: Timer Value (rsvd_tmr_val):** Reference only: This read-only field returns the running count of the PM timer. This counter runs off a 3.579545 MHz clock (derived from 14.31818 MHz divided by 4). It is reset (to 0) during a Platform reset, and then continues counting as long as the system is in the S0 state. Anytime bit 22 of the timer goes HIGH to LOW (bits referenced from 0 to 23), the TMROF_STS bit is set. The High-to-Low transition will occur every 2.3435 seconds. Writes to this register have no effect.

10.63 General Purpose Event 0 Status (GPE0a_STS)—Offset 20h

Note: This register is symmetrical to the General Purpose Event 0a Enable Register. Unless indicated otherwise below, if the corresponding _EN bit is set, then when the STS bit get set, the PMC will generate a Wake Event. Once back in an S0 state (or if already in an S0 state when the event occurs), the PMC will also generate an SCI if the SCI_EN bit is set, or an SMI# if the SCI_EN bit is not set.

Access Method

Type: IO Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (rsvd31):** Reserved.
21	0h RW/1C/V	**CSE SCI Status (cse_sci_sts):** This bit will be set to 1 by the PMC when CSE sends an Assert_SCI message to the PMC. Additionally, if the CSE_SCI_EN and SCI_EN bits are set, and the system is in an S0 state, then the setting of the CSE_SCI_STS bit will generate an SCI (or SMI# if SCI_EN is not set but GLB_SMI_EN is). This bit is cleared by a software write of '1'.
20	0h RW/1C/V	**eSPI SCI Status (espi_sci_sts):** This bit will be set when an agent attached to eSPI is requesting an SCI via an Assert_SCI message with tag 001b. Note: This source is not able to cause a SX wake event.
19	0h RW/1C/V	**SPI PME Status (spi_pme_sts):** This bit will be set to 1 by the PMC when eSPI sends a PME Virtual Wire message to the PMC with VWIDX==12h, VWVLD[3]==1, VWLVL[3]==0 (PME#). Additionally, if the SPI_PME_EN and SCI_EN bits are set, and the system is in an S0 state, then the setting of the SPI _PME_STS bit will generate an SCI (or SMI# if SCI_EN is not set but GLB_SMI_EN is). This bit is cleared by a software write of '1'.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW/1C/V	**CNV PME Status (cnv_pme_sts):** This bit will be set to 1 by the PMC when CNV sends Assert_PME message to PMC as a result of a wake event from Wi-Fi in D3. Additionally, if the CNV_PME_EN and SCI_EN bits are set, and the system is in an S0 state, then the setting of the CNV_PME_STS bit will generate an SCI (or SMI# if SCI_EN is not set but GLB_SMI_EN is). This bit is cleared by a software write of '1'.
17	0h RW/1C/V	**SATA PME Status (sata_pme_sts):** This bit will be set to 1 by the PMC when SATA asserts its PME wire signal. Additionally, if the SATA_PME_EN and SCI_EN bits are set, and the system is in an S0 state, then the setting of the SATA _PME_STS bit will generate an SCI (or SMI# if SCI_EN is not set but GLB_SMI_EN is). This bit is cleared by a software write of '1'.
16	0h RW/1C/V	**SMBUS Wake Status (smb_wak_sts):** This bit is set to 1 by the hardware to indicate that the wake event was caused by the PCH's SMBus logic (SMBALERT# signal going active). Notes: 1. The SMBus controller will independently cause an SMI message so this bit does not need to do so (unlike the other bits in this register). 2. This bit is set by the SMBus slave command 01h (Wake/SMI#) even when the system is in the S0 state. Therefore, to avoid an instant wake on subsequent transitions to sleep states, software must clear this bit after each reception of the Wake/SMI# command or just prior to entering the sleep state. 3. The SMBALERT_STS bit (D31:F3:I/O Offset 00h:bit 5) should be cleared.
15	0h RW/1C/V	**GPIO Tier1 SCI Status (gpio_tier1_sci_sts):** This bit is a logical OR of sci_wake wires from tier 1 GPIO's. Additionally, if the GPIO_TIER1_SCI_EN and SCI_EN bits are set, and the system is in an S0 state, then the setting of the GPIO_TIER1_SCI_STS bit will generate an SCI (or SMI# if SCI_EN is not set but GLB_SMI_EN is). This bit is cleared by a software write of '1'. If the GPIO_TIER1_SCI_EN is set this bit is a SX and S0ix wake source. There may be additional PMU wake mask bits for SX and SOix wakes.
14	0h RW/1C/V	**AVS PME Status (avs_pme_sts):** This bit will be set to 1 by the PMC when AVS asserts its PME wire signal. Additionally, if the AVS_PME_EN and SCI_EN bits are set, and the system is in an S0 state, then the setting of the AVS_PME_STS bit will generate an SCI (or SMI# if SCI_EN is not set but GLB_SMI_EN is). This bit is cleared by a software write of '1'.
13	0h RW/1C/V	**USB XHCI PME Status (xhci_pme_sts):** This bit will be set to 1 by the PMC when USB XHCI asserts its PME wire signal. Additionally, if the XHCI_PME_EN and SCI_EN bits are set, and the system is in an S0 state, then the setting of the XHCI_PME_STS bit will generate an SCI (or SMI# if SCI_EN is not set but GLB_SMI_EN is). This bit is cleared by a software write of '1'.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW/1C/V	**USB XDCI PME Status (xdci_pme_sts):** This bit will be set to 1 by the PMC when USB XDCI asserts its PME wire signal. Additionally, if the XDCI_PME_EN and SCI_EN bits are set, and the system is in an S0 state, then the setting of the XDCI_PME_STS bit will generate an SCI (or SMI# if SCI_EN is not set but GLB_SMI_EN is). This bit is cleared by a software write of '1'.
11	0h RW/1C/V	**CSE PME Status (cse_pme_sts):** This bit will be set to 1 by the PMC when CSE sends Assert_PME message to PMC. Additionally, if the CSE_PME_EN and SCI_EN bits are set, and the system is in an S0 state, then the setting of the CSE_PME_STS bit will generate an SCI (or SMI# if SCI_EN is not set but GLB_SMI_EN is). This bit is cleared by a software write of '1'.
10	0h RW/1C/V	**Battery Low Status (batlow_sts):** This bit will be set to 1 by hardware when the PMU_BATLOW_B signal goes low. Software clears the bit by writing a 1 to the bit position. In Desktop Mode, this bit will be treated as Reserved.
9	0h RW/1C/V	**PCIE GPE Status (pcie_gpe_sts):** This bit will be set to 1 by hardware to indicate that the PMC received an Assert_GPE message from one or more of the PCI-Express Ports. The GPE messages represents a PME event message from one or more of the PCI-Express Ports. Software attempts to clear this bit by writing a 1 to this bit position. If the bit is not cleared and the corresponding PCI_EXP_EN bit is set, the level triggered SCI will remain active. Note that a race condition exists where the PCI Express device sends another PME message because the PCI Express device was not serviced within the time when it must resend the message. This may result in a spurious interrupt, and this is comprehended and approved by the PCI Express specification. The window for this race condition is approximately 95-105 milliseconds.
8	0h RW/1C/V	**PCI Express Wake3 Status (pcie_wake3_sts):** This bit is set by hardware to indicate that the pci_wake3_b pin was asserted. Software writes a 1 to clear this bit. If pci_wake3_b pin is still active during the write, then the bit will remain active (i.e. input to this bit is level sensitive).
7	0h RW/1C/V	**PCI Express Wake2 Status (pcie_wake2_sts):** This bit is set by hardware to indicate that the pci_wake2_b pin was asserted. Software writes a 1 to clear this bit. If pci_wake2_b pin is still active during the write, then the bit will remain active (i.e. input to this bit is level sensitive).
6	0h RW/1C/V	**PCI Express Wake1 Status (pcie_wake1_sts):** This bit is set by hardware to indicate that the pci_wake1_b pin was asserted. Software writes a 1 to clear this bit. If pci_wake1_b pin is still active during the write, then the bit will remain active (i.e. input to this bit is level sensitive).
5	0h RO	**Reserved (rsvd5):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW/1C/V	**PUNIT SCI status (punit_sci_sts):** This bit will be set if the Power Management Unit requests SCI or and SMI if PUNIT_SCI_EN and !SCI_EN and GBL_SMI_EN is set. This bit is cleared by writing a 1 to this bit position.
3	0h RW/1C/V	**PCI Express Wake0 Status (pcie_wake0_sts):** This bit is set by hardware to indicate that the pci_wake0_b pin was asserted. Software writes a 1 to clear this bit. If pci_wake0_b pin is still active during the write, then the bit will remain active (i.e. input to this bit is level sensitive).
2	0h RW/1C/V	**Software GPE Status (swgpe_sts):** The SWGPE_CTRL bit (bit 1 of GPE_CTRL reg) acts as a level input to this bit.
1	0h RO	**Reserved (rsvd):** Reserved.
0	0h RW/1C/V	**PCIE SCI (pcie_sci_sts):** This bit will be set to 1 by the PMC when Pcie sends Assert_SCI message to PMC. Additionally, if the pcie_sci_en and sci_en bits are set, and the system is in an S0 state, then the setting of the pcie_sci_sts bit will generate an SCI (or SMI# if sci_en is not set but GLB_SMI_EN is). This bit is cleared by a software write of '1'. Note: TAG field of Assert_SCI message is ignored.

10.64 General Purpose Event 0 Status (GPE0b_STS)—Offset 24h

Note: This register is symmetrical to the General Purpose Event 0b Enable Register. Reads/writes to this register will result in the transaction being forwarded to the corresponding GPIO community based on the GPIO_GPE_CFG.gpe0_dw1 register configuration.

Access Method

Type: IO Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**General Purpose Event 0 Status Dword 1 (gpe0b_sts):** Shadow of the general purpose event status from the corresponding GPIO community specified in GPIO_GPE_CFG.gpe0_dw1.

10.65 General Purpose Event 0 Status (GPE0c_STS)—Offset 28h

Note: This register is symmetrical to the General Purpose Event 0c Enable Register. Reads/writes to this register will result in the transaction being forwarded to the corresponding GPIO community based on the GPIO_GPE_CFG.gpe0_dw2 register configuration.

Access Method

Type: IO Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**General Purpose Event 0 Status Dword 2 (gpe0c_sts):** Shadow of the general purpose event status from the corresponding GPIO community specified in GPIO_GPE_CFG.gpe0_dw2.

10.66 General Purpose Event 0 Status (GPE0d_STS)—Offset 2Ch

Note: This register is symmetrical to the General Purpose Event 0d Enable Register. Reads/writes to this register will result in the transaction being forwarded to the corresponding GPIO community based on the GPIO_GPE_CFG.gpe0_dw3 register configuration.

Access Method

Type: IO Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**General Purpose Event 0 Status Dword 3 (gpe0d_sts):** Shadow of the general purpose event status from the corresponding GPIO community specified in GPIO_GPE_CFG.gpe0_dw3.

10.67 General Purpose Event 0 Enables (GPE0a_EN)—Offset 30h

Note: This register is symmetrical to the General Purpose Event 0a Status Register.

Access Method

Type: IO Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (rsvd31):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**CSE SCI Enable (cse_sci_en):** Enables the setting of the CSE_SCI STS bit to generate a wake event and/or an SCI or SMI#.
20	0h RW	**eSPI SCI Enable (espi_sci_en):** Enables the setting of the ESPI_SCI STS bit to generate a wake event and/or an SCI or SMI#.
19	0h RW	**SPI PME Enable (spi_pme_en):** Enables the setting of the SPI_PME_STS bit to generate a wake event and/or an SCI or SMI#. Additionally on those platforms where this bit affects SX wake events, the PMC must clear this field on emergency shutdowns like powerbutton override and CPU thermal trip
18	0h RW	**CNV PME Enable (cnv_pme_en):** Enables the setting of the CNV_PME_STS bit to generate a wake event and/or an SCI or SMI#.
17	0h RW	**SATA PME Enable (sata_pme_en):** Enables the setting of the SATA_PME_STS bit to generate a wake event and/or an SCI or SMI#.
16	0h RW/V	**SMBUS WAKE Enable (smb_wak_en):** Enables the setting of the SMB_WAK_STS bit to generate a wake event BUT NOT an SCI or SMI#. SMB_WAK is a special case in that it cannot cause an SCI or SMI. On those platforms where this bit affects SX wake events, the PMC must clear this field on emergency shutdowns like powerbutton override and CPU thermal trip.
15	0h RW	**GPIO Tier1 SCI Enable (gpio_tier1_sci_en):** Enables the setting of the GPIO_TIER1_SCI_STS bit to generate a wake event and/or an SCI or SMI#. On those platforms where this bit affects SX wake events, the PMC must clear this field on emergency shutdowns like powerbutton override and CPU thermal trip.
14	0h RW	**AVS PME Enable (avs_pme_en):** Enables the setting of the AVS PME STS bit to generate a wake event and/or an SCI or SMI#. On those platforms where this bit affects SX wake events, the PMC must clear this field on emergency shutdowns like powerbutton override and CPU thermal trip.
13	0h RW	**USB XHCI PME Enable (xhci_pme_en):** Enables the setting of the XHCI_PME_STS bit to generate a wake event and/or an SCI or SMI#.
12	0h RW	**USB XDCI PME Enable (xdci_pme_en):** Enables the setting of the XDCI_PME_STS bit to generate a wake event and/or an SCI or SMI#.
11	0h RW	**CSE PME Enable (cse_pme_en):** Enables the setting of the CSE_PME_EN bit to generate a wake event and/or an SCI or SMI#.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW	**Battery Low Enable (batlow_en):** This bit enables the PMU_BATLOW_B signal to cause an SMI# or SCI (depending on the SCI_EN bit) when it goes low.This bit does not prevent the PMU_BATLOW_B signal from inhibiting the wake event. On desktop platforms this pin should be tied high to ensure proper functionality. In addition to its normal reset conditions, the PMC also clears this bit on emergency shutdowns like power button override and CPU thermal trip.
9	0h RW	**PCIE GPE Enable (pcie_gpe_en):** Enables the PMC to cause an SCI or SMI when the PCIE_GPE_STS bit is set. This is used to allow the PCI Express ports to cause an SCI or SMI due to pmc receiving an Assert_GPE from PCIE0 or PCIE1.
8	0h RW/V	**PCI Express Wake3 Enable (pcie_wake3_en):** This bit, when set to 1, enables the PCIE_WAKE3_STS to to cause an SCI. On those platforms where this bit affects SX wake events, the PMC must clear this field on emergency shutdowns like powerbutton override and CPU thermal trip.
7	0h RW/V	**PCI Express Wake2 Enable (pcie_wake2_en):** This bit, when set to 1, enables the PCIE_WAKE2_STS to to cause an SCI. On those platforms where this bit affects SX wake events, the PMC must clear this field on emergency shutdowns like powerbutton override and CPU thermal trip.
6	0h RW/V	**PCI Express Wake1 Enable (pcie_wake1_en):** This bit, when set to 1, enables the PCIE_WAKE1_STS to to cause an SCI. On those platforms where this bit affects SX wake events, the PMC must clear this field on emergency shutdowns like powerbutton override and CPU thermal trip.
5	0h RO	**Reserved (rsvd5):** Reserved.
4	0h RW	**PUnit SCI Enable (punit_sci_en):** This bit enables the corresponding PUNIT_SCI_STS bit being set to cause an SCI and/or wake event.
3	0h RW/V	**PCI Express Wake0 Enable (pcie_wake0_en):** This bit, when set to 1, enables the PCIE_WAKE0_STS to to cause an SCI. On those platforms where this bit affects SX wake events, the PMC must clear this field on emergency shutdowns like powerbutton override and CPU thermal trip.
2	0h RW	**Software GPE Enable (swgpe_en):** This bit, when set to 1, enables the SW GPE function. If SWGPE_CTRL is written to a 1, hardware will set SWGPE_STS (acts as a level input). If SWGPE_STS, SWGPE_EN, and SCI_EN are all 1's, an SCI will be generated. If SWGPE_STS = 1, SWGPE_EN = 1, SCI_EN = 0, and GBL_SMI_EN = 1 then an SMI# will be generated.
1	0h RO	**Reserved (rsvd):** Reserved.
0	0h RW	**PCIE SCI Enable (pcie_sci_en):** Enables the setting of the PCIE_SCI_EN bit to generate a wake event and/or an SCI or SMI#.

10.68 General Purpose Event 0 Enable (GPE0b_EN)—Offset 34h

Note: This register is symmetrical to the General Purpose Event 0b Status Register. Reads/writes to this register will result in the transaction being forwarded to the corresponding GPIO community based on the GPIO_GPE_CFG.gpe0_dw1 register configuration.

Access Method

Type: IO Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**General Purpose Event 0 Enable Dword 1 (gpe0b_en):** Shadow of the general purpose event enable from the corresponding GPIO community specified in GPIO_GPE_CFG.gpe0_dw1.

10.69 General Purpose Event 0 Enable (GPE0c_EN)—Offset 38h

Note: This register is symmetrical to the General Purpose Event 0c Status Register. Reads/writes to this register will result in the transaction being forwarded to the corresponding GPIO community based on the GPIO_GPE_CFG.gpe0_dw2 register configuration.

Access Method

Type: IO Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**General Purpose Event 0 Enable Dword 2 (gpe0c_en):** Shadow of the general purpose event enable from the corresponding GPIO community specified in GPIO_GPE_CFG.gpe0_dw2.

10.70 General Purpose Event 0 Enable (GPE0d_EN)—Offset 3Ch

Note: This register is symmetrical to the General Purpose Event 0d Status Register. Reads/writes to this register will result in the transaction being forwarded to the corresponding GPIO community based on the GPIO_GPE_CFG.gpe0_dw3 register configuration.

Access Method

Type: IO Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**General Purpose Event 0 Enable Dword 3 (gpe0d_en):** Shadow of the general purpose event enable from the corresponding GPIO community specified in GPIO_GPE_CFG.gpe0_dw3.

10.71 SMI Control and Enable (SMI_EN)—Offset 40h

Access Method

Type: IO Register
(Size: 32 bits)

Device:
Function:

Default: 2h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (rsvd):** Reserved.
29	0h RW	**CSE SMI Enable (cse_smi_en):** Enables SMI logic to cause SMI if gbl_smi_en is set.
28	0h RW/L	**eSPI SMI Enable (espi_smi_en):** Enables SMI logic to cause SMI if gbl_smi_en is set. Software sets this bit to enable eSPI SMI events. NOTE: When the GCR.LOCK.ESPI_SMI bit is set, this bit cannot be changed.
27	0h RW	**PMC OCP Fabric SMI Enable (ocp_smi_en):** Enables SMI logic to cause SMI if gbl_smi_en is set.
26	0h RW	**SPI SMI Enable (spi_smi_en):** Enables SMI logic to cause SMI if gbl_smi_en is set.
25	0h RO	**Reserved (rsvd25):** Reserved.
24:22	0h RO	**Reserved (reserved24_22):** Reserved.
21	0h RW	**SCC2 SMI Enable (scc2_smi_en):** Enables SMI logic to cause SMI if gbl_smi_en is set.
20	0h RW	**PCIE SMI Enable (pcie_smi_en):** Enables SMI logic to cause SMI if gbl_smi_en is set.
19	0h RW	**SCS SMI Enable (scs_smi_en):** Enables SMI logic to cause SMI if gbl_smi_en is set.
18	0h RW	**Host SMBUS SMI Enable (host_smbus_smi_en):** Enables SMI logic to cause SMI if gbl_smi_en is set.

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**XHCI SMI Enable (xhci_smi_en):** Enables SMI logic to cause SMI if gbl_smi_en is set.
16	0h RW	**SMBUS SMI Enable (smbus_smi_en):** Enables SMI logic to cause SMI if gbl_smi_en is set.
15	0h RW	**SERIRQ SMI Enable (serirq_smi_en):** Enables SMI logic to cause Sync SMI if gbl_smi_en is set.
14	0h RW	**Periodic Enable (periodic_en):** Setting this bit will cause the PMC to generate an SMI# when the PERIODIC_STS bit is set in the SMI_STS register.
13	0h RW/L	**TCO Enable (tco_en):** When set, enables the TCO logic to generate SMI#. When cleared, disables TCO logic generating an SMI#. Note: This bit cannot be written once the TCO_LOCK bit is set. This prevents unauthorized software from disabling the generation of TCO-based SMIs.
12	0h RW	**MCSMI Enable (mcsmi_en):** Software sets this bit to 1 to enables Intel PCH to trap access to the microcontroller range (62h or 66h). A 'trapped' cycles will be claimed by Intel PCH, but not forwarded to LPC/eSPI. An SMI# will also be generated.
11	0h RW/1S	**GPIO Unlock SSMI EN (gpio_unlock_ssmi_en):** Enables SSMI logic to cause SSMI if gbl_smi_en is set and parallel STS bit is set. Once written to '1', this bit can only be cleared by the reset condition for this bit.
10	0h RW	**GPIO SMI Enable (gpio_smi_en):** Enables SMI logic to cause SMI if gbl_smi_en is set.
9:8	0h RO	**Reserved (reserved5):** Reserved.
7	0h WO	**BIOS RLS (bios_rls):** Enables the generation of an SCI interrupt for ACPI software when a one is written to this bit position by BIOS software. This bit always reads a zero. Note: GBL_STS being set will cause an SCI, even if the SCI_EN bit is not set. Software must take great care not to set the BIOS_RLS bit (which causes GBL_STS to be set) if the SCI handler is not in place.
6	0h RW	**Software SMI Timer Enable (swsmi_tmr_en):** Software sets this bit to a 1 to start the Software SMI# Timer. When the timer expires (depending on the SWSMI_RATE_SEL bits), it will generate an SMI# and set the SWSMI_TMR_STS bit. The SWSMI_TMR_EN bit will remain at 1 until software sets it back to 0. Clearing the SWSMI_TMR_EN bit before the timer expires will reset the timer and the SMI# will not be generated. The default for this bit is 0.
5	0h RW	**APMC Enable (apmc_en):** If set, this enables writes to the APM register to cause an SMI#.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW	**SMI On Sleep Enable (smi_on_slp_en):** If this bit is set, the PMC will generate an SMI# when a write access attempts to set the SLP_EN bit (in the PM1_CNT register). Furthermore, the PMC will not put the system to a sleep state. This allows the SMI# handler work around chip-level bugs. It is expected that the SMI# handler will turn off the SMI_ON_SLP_EN bit before actually setting the SLP_EN bit.
3	0h RW	**Legacy USB Enable (legacy_usb_en):** Enables SMI logic to cause SMI if gbl_smi_en is set.
2	0h RW	**BIOS Enable (bios_en):** Enables the generation of SMI# when ACPI software writes a 1 to the GBL_RLS bit. Note that if the BIOS_STS bit, which gets set when software writes a 1 to GBL_RLS bit, is already a 1 at the time that BIOS_EN becomes 1, an SMI# will be generated when BIOS_EN gets set.
1	1h RW/1S/V	**End of SMI (eos):** SMI handler sets this bit when it finishes handling the SMI. Setting this bit will force the PMC internal SMI request to zero for 1 clock.Thus if the internal SMI request is still asserted due to new SMI trigger and SMI_ACK was received for the previous SMI, a new SMI will be sent. Hardware clears this bit, SW/FW sets.
0	0h RW/L	**Global SMI Enable (gbl_smi_en):** When set, this bit enables the generation of SMIs in the system upon any enabled SMI event. If this bit is not set, no SMI# will be generated. Note: When the SMI_LOCK bit is set, this bit cannot be changed.

10.72 SMI Status Register (SMI_STS)—Offset 44h

Access Method

Type: IO Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (rsvd):** Reserved.
29	0h RW/1C/V	**CSE SMI Status (cse_smi_sts):** This bit will be set if CSE is requesting an SMI#. Writing a '1' to this bit position clears this bit to '0'.
28	0h RW/1C/V	**eSPI SMI Status (espi_smi_sts):** This bit is set if an eSPI agent is requesting an SMI# via an Assert_SMI message with tag 100b. This bit is set by hardware and cleared by software writing a 1 to its bit position.
27	0h RW/1C/V	**PMC OCP Fabric Status (ocp_smi_sts):** This bit will be set if PMC OCP fabric (via PMC IOSF2OCP bridge) requests an SMI#. Writing a '1' to this bit position clears this bit to '0'.

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW/1C/V	**SPI SMI Status (spi_smi_sts):** This bit will be set when the SPI logic is requesting an SMI# via a message sent from SPI to PMC# with tag 111b.
25	0h RO	**Reserved (rsvd25):** Reserved.
24:22	0h RO	**Reserved (reserved2):** Reserved.
21	0h RW/1C/V	**SCC2 SMI Status (scc2_smi_sts):** This bit will be set if SCC2 unit is requesting an SMI#. Writing a '1' to this bit position clears this bit to '0'.
20	0h RW/1C/V	**PCIE_SMI Status (pcie_smi_sts):** PCIE SMI event occurred. Set with Assert_SMI from PCIE0 or PCIE1. Writing a '1' to this bit position clears this bit to '0'
19	0h RW/1C/V	**SCS SMI Status (scs_smi_sts):** This bit will be set if SCS unit is requesting an SMI#. Writing a '1' to this bit position clears this bit to '0'.
18	0h RW/1C/V	**Host SMBUS SMI Status (host_smbus_smi_sts):** The PMC sets this bit to 1 to indicate that the SMI# was triggered when Host initiated SMBUS transaction encounters failure such as device error or bus collision or upon the completion of a valid transaction on the bus but with SMBUS SIP SMB_SMI_EN = 1. Further clarification of the cause of this SMI should be queried from SMBUS IPs HOST STATUS register. PMC sets this bit on receiving an SMI message with tag 0x1.
17	0h RW/1C/V	**USB XHCI SMI Status (xhci_smi_sts):** This bit will be set when the USB logic is requesting an SMI#.
16	0h RW/1C/V	**SMBUS SMI Status (smbus_smi_sts):** The PMC sets this bit to 1 to indicate that the SMI# was caused by: 1. The SMBUS Slave receiving a message that an SMI# should be caused. This implies that the PMC received an SMI message from SMBUS with tag 0x3. 2. The SMBALERT# signal goes active and the SMB_SMI_EN bit is set and the SMBALERT_DIS bit is cleared. This implies that the PMC received an SMI message from SMBUS with tag 0x2. 3. The SMBUS Slave receiving a HOST_NOTIFY message and the HOST_NOTIFY_INTREN and the SMB_SMI_EN bits are set. Where HOST_NOTIFY_INTREN is available via the SMBUS IP not in the PMC. This implies that the PMC received an SMI message from SMBUS with tag 0x2. 4. The SMBUS Slave receiving a SMI in S0 message. This implies that the PMC received an SMI message from SMBUS with tag 0x2. This bit is sticky. It is cleared by writing a 1 to this bit position. Note that this bit is set from the 64 KHz clock domain used by the SMBUS. Software must wait at least 15.63 microseconds

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C/V	**SERIRQ SMI Status (serirq_smi_sts):** 1: Indicates the SMI# was caused by the SERIRQ decoder. This implies that the PMC received an SMI message with tag 011b from eSPI or an SMI message with any tag from LPC. 0: SMI# not caused by SERIRQ decoder. This bit will remain 1 until the software writes a 1 to this bit.
14	0h RW/1C/V	**Periodic Status (periodic_sts):** This bit will be set at the rate determined by the PER_SMI_SEL bits. If the PERIODIC_EN bit is also set, the PMC will generate an SMI#. This bit is cleared by writing a 1 to this bit position.
13	0h RW/1C/V	**TCO Status (tco_sts):** Indicates SMI was caused by the TCO logic. This bit is cleared by writing a 1 to this bit position, Causes would be: 1. Century rollover from SMBUS. This functionality comes from the RTC IP. 2. TCO_TMR 1st & 2nd expiration in PMC. 3. OS writes to TCO_DAT_IN register. Sent to PMC as a SMI message with tag 0x0 from SMBUS. 4. NMI occurred and mapped to SMI. Sent to PMC as a SMI message with not tag from ITSS. 5. INTRUDER# signal goes active. Wire to the PMC from SMBUS. 6. Illegal attempt to write to BIOS located in the FWH accessed over LPC. Sent to PMC as a SMI msg with tag 0x0 from SMBUS. 7. Changes of BC.WPD (Write Protect Disable) bit from 0 to 1. Sent to PMC as an SyncSMI message with tag 0x1 from LPC or eSPI. 8. BIOS Write Protect Disable (SPI_BC.WPD) set if BIOS Lock Enable (SPI_BC.LE) = 1 Sent to PMC as an SyncSMI message with tag 0x5 from SPI
12	0h RW/1C/V	**Microcontroller SMI Status (mcsmi_sts):** This bit is set if there is an access to the power management microcontroller range (62h or 66h). If this bit is set, and the MCSMI_EN bit is also set, the Intel PCH will generate an SMI#. This bit is set by hardware and cleared by software writing a 1 to its bit position. This bit will be set when the eSPI/LPC logic is requesting an Sync SMI# with tag 0x0. The FW controlled ACPI_CTL.MCSMI_CTRL bit acts as a rising edge input to this bit.
11	0h RW/1C/V	**GPIO Unlock SSMI STS (gpio_unlock_ssmi_sts):** This bit will be set if the GIO registers lockdown logic is requested an SSMI. Writing a '1' to this bit position clears this bit to '0'.
10	0h RW/1C/V	**GPIO SMI STS (gpio_smi_sts):** This bit will be a 1 if any GPIO that is enabled to trigger SMI is asserted. GPIOs that are not routed to cause an SMI will have no effect on this bit.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RO	**Reserved (rsvd_gpe0_sts):** GPE0a Status Register. Historically GPE0a_STS could only generate SCI and never an SMI. Reference only: There are several status/enable bit pairs in GPE0a_STS/EN that are capable of triggering SMI. This bit is a logical OR of all of those pairs (i.e. this bit is asserted whenever at least one of those pairs has both the status and enable bit asserted). This bit is NOT sticky. Writes to this bit will have no effect. Note: The setting of this bit does not cause the SMI#.
8	0h RO	**Reserved (rsvd_pm1_sts_reg):** PM1 Status Register. Historically, PM1_STS_EN could only generate SCI and never an SMI. Reference only: This is an OR of the bits (except for bits 5 and 4) in the ACPI PM1_STS_EN Status Reg. Not sticky. Writes to this bit have no effect. Note: The setting of this bit does not cause the SMI#.
7	0h RO	**Reserved (reserved4):** Reserved. No corresponding status bit to BIOS_RLS.
6	0h RW/1C/V	**Software SMI Timer Status (swsmi_tmr_sts):** This bit will be set to 1 by the hardware when the Software SMI# Timer expires. This bit will remain 1 until the software writes a 1 to this bit.
5	0h RW/1C/V	**Advanced Power Management Status (apm_sts):** SMI# was generated by a write access to the APM control register and if the APMC_EN bit is set. This bit is cleared by writing a one to its bit position.
4	0h RW/1C/V	**SMI ON SLP EN Status (smi_on_slp_en_sts):** This bit will be set by the PMC when a write access attempts to set the SLP_EN bit. This bit is cleared by writing a 1 to this bit position.
3	0h RW/1C/V	**Legacy USB Status (legacy_usb_sts):** Logical OR of each of the SMI status bits in the USB Legacy Keybd Register in the eSPI or LPC IP ANDed with the corresponding enable bits in the LPC IP. SMI_EN.LEGACY_USB_EN must be set for this bit to cause an SMI. This bit is cleared by writing a 1 to this bit position. This bit will be set when the eSPI or LPC logic is requesting a Sync SMI# with tag 0x2. The FW controlled ACPI_CTL.LEGACY_USB_CTRL bit acts as a rising edge input to this bit.
2	0h RW/1C/V	**BIOS Status (bios_sts):** This bit gets set by hardware when a 1 is written by software to the GBL_RLS bit. When both BIOS_EN and the BIOS_STS bit are set, an SMI# will be generated. The BIOS_STS bit is cleared when software writes a 1 to this bit position.
1:0	0h RO	**Reserved (reserved7):** Reserved.

10.73 Device Trap Status (DEVTRAP_STS)—Offset 4Ch

Each bit indicates if an access has occurred to the corresponding device's trap range or for bits 6:9 if the corresponding PCI interrupt is active. Write 1 to the same bit position to clear it. This register is used by APM power management software to see if there has been system activity. The periodic SMI# timer indicates if it is the right time to read the DEVTRAP_STS register.

Access Method

Type: IO Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (rsvd16):** Reserved.
15:13	0h RO	**Reserved (rsvd15):** Reserved.
12	0h RW/1C/V	**Device Trap Status Bit 12 (d12_trp_sts):** KBC (60/64h)
11:6	0h RO	**Reserved (rsvd11):** Reserved.
5	0h RW/1C/V	**Device Trap Status Bit 5 (d5_trp_sts):** This will be set if any of the following are accessed (as determined by the I/O ranges in the LPC decoder (even if the LPC forwarding is not enabled): SP1, SP2, PP, FDC.
4:0	0h RO	**Reserved (rsvd4):** Reserved.

10.74 General Purpose Event Control (GPE_CTRL)—Offset 50h

Access Method

Type: IO Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Reserved (rsvd):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**Software General Purpose Event Control (swgpe_ctrl):** This bit allows software to control the assertion of SWGPE_STS bit. This bit is used by hardware as the level input signal for the SWGPE_STS bit in the GPE0a_STS register. When SWGPE_CTRL is 1, SWGPE_STS will be set to 1, and writes to SWGPE_STS with a value of 1 to clear SWGPE_STS will result in SWGPE_STS being set back to 1 by hardware. When SWGPE_CTRL is 0, writes to SWGPE_STS with a value of 1 will clear SWGPE_STS to 0.
16:0	0h RO	**Reserved (reserved1):** Reserved.

10.75 TCO Reload Register (TCO_RLD)—Offset 60h

Access Method

Type: IO Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RO	**Reserved (rsvd):** Reserved.
9:0	0h RO/V	**TCO Timer Value (tco_val):** Reading this register will return the current count of the TCO timer. Writing any value to this register will reload the timer to prevent the timeout.

10.76 TCO Timer Status (TCO_STS)—Offset 64h

Access Method

Type: IO Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Reserved (reserved2):** Reserved.
17	0h RW/1C/V	**Second Timeout Status (second_to_sts):** PMC sets this bit to 1 to indicate that the TIMEOUT bit had been (or is currently) set and a second timeout occurred before the TCO_RLD register was written. If this bit is set and the NO_REBOOT config bit is 0, then the PMC will reboot the system after the second timeout. The reboot is done by interrupting the Arc and starting a reset flow based on the OS_POLICY. This bit is only cleared by writing a 1 to this bit or by a reset.

Bit Range	Default & Access	Field Name (ID): Description
16:4	0h RO	**Reserved (reserved1):** Reserved.
3	0h RW/1C/V	**TCO Timeout (tco_timeout):** Bit set to 1 by PMC to indicate that the SMI was caused by TCO timer reaching 0.
2:0	0h RO	**Reserved (rsvd):** Reserved.

10.77 TCO Timer Control (TCO1_CNT)—Offset 68h

Access Method

Type: IO Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (rsvd):** Reserved.
21:20	0h RW	**OS Policy (os_policy):** OS-based software writes to these bits to select the policy that the BIOS will use after the platform resets due the WDT. The following convention is recommended for the BIOS and OS: 00: Boot normally 01: Shut down 10: Don't load OS. Hold in pre-boot state and use LAN to determine next step. 11: Reserved
19:13	0h RO	**Reserved (reserved2):** Reserved.
12	0h RW/L	**TCO Lock (tco_lock):** When set to 1, this bit prevents writes from changing the TCO_EN bit (in offset 30h of Power Management I/O space). Once this bit is set to 1, it cannot be cleared by software writing a 0 to this bit location. Reset is required to change this bit from 1 to 0. This bit defaults to 0. On some prior platforms, this field is reset on cold reset.
11	0h RW	**TCO Timer Halt (tco_tmr_halt):** 1: The TCO timer will halt. It will not count, and thus cannot reach a value that would cause an SMI# or to cause the SECOND_TO_STS bit to be set. This will also prevent rebooting. 0: The TCO timer is enabled to count. This is the default.
10:0	0h RO	**Reserved (reserved1):** Reserved.

10.78 TCO Timer Register (TCO_TMR)—Offset 70h

Access Method

Type: IO Register
(Size: 32 bits)

Device:
Function:

Default: 40000h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (reserved1):** Reserved.
25:16	4h RW	**TCO Timer Reload Value (tco_trld_val):** Value that is loaded into the timer each time the TCO_RLD register is written. Values of 0000h or 0001h will be ignored and should not be attempted. The timer is clocked at approximately 0.6 seconds, and thus allows timeouts ranging from 1.2 second to 613.8 seconds. Note: The timer has an error of +/- 1 tick (0.6s). The TCO Timer will only count down in the S0 and S0IX state.
15:0	0h RO	**Reserved (rsvd):** Reserved.

10.79 Advanced Power Management Status (APM_STS)—Offset 74h

Access Method

Type: IO Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**Advanced Power Management Status Port (apm_sts):** Used to pass data between the OS and the SMI handler. Basically, this is a scratch pad register and is not effected by any other register or function (other reset). Writes can come through ACPI register writes or from IO Port write to port 0xB3.
23:0	0h RO	**Reserved (rsvd):** Reserved.

10.80 Advanced Power Management Control Port (APM_CNT)—Offset 78h

Access Method

Type: IO Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (reserved0):** Reserved.
23:16	0h RW	**Advanced Power Management Control Port (apm_cnt):** Used to pass an APM command between the OS and the SMI handler. Writes to this bit field not only store data in the APMC register, but also generates an SMI# when the APMC_EN bit is set. Writes can come through ACPI register writes or from IO Port write to port 0xB2.
15:0	0h RO	**Reserved (reserved1):** Reserved.

10.81 Direct IRQ Enables (DIRECT_IRQ_EN)—Offset 7Ch

Access Method

Type: IO Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RO	**Reserved (reserved0):** Reserved.
9	0h RW	**Direct IRQ Enable for PMIC (pmic_en):** This bit enables generation of De/AssertIRQn message based on SVID's interrupt signal (which reflects PMIC's alert signal). The resulting De/AssertIRQn will have an IRQ vector, as programmed in the corresponding GCR.IRQ_SEL_0/1/2 field.
8	0h RW	**Direct IRQ Enable for XHCI (xhci_en):** This bit enables wake from S0ix and generation of De/AssertIRQn message based on XHCI's PME & D0i3 signals. The resulting De/AssertIRQn will have vector. as programmed in the corresponding GCR.IRQ_SEL_0/1 field.
7	0h RW	**Direct IRQ Enable for XDCI (xdci_en):** This bit enables wake from S0ix and generation of De/AssertIRQn message based on XDCI's PME & D0i3 signals. The resulting De/AssertIRQn will have vector. as programmed in the corresponding GCR.IRQ_SEL_0/1 field.
6	0h RW	**Direct IRQ Enable for SCC (sdio_d1_en):** This bit enables wake from S0ix and there is no De/AssertIRQn message for SDIO_D1 only a wake.
5	0h RW	**Direct IRQ Enable for SCC (sdcard_cd_en):** This bit enables wake from S0ix and generation of De/AssertIRQn message based on SCC's PME & D0i3 signals. The resulting De/AssertIRQn will have vector. as programmed in the corresponding GCR.IRQ_SEL_0/1 field.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW	**Direct IRQ Enable for SCC (sdcard_d1_en):** This bit enables wake from S0ix and generation of De/AssertIRQn message based on SCC's PME & D0i3 signals. The resulting De/AssertIRQn will have vector. as programmed in the corresponding GCR.IRQ_SEL_0/1 field.
3	0h RW	**Direct IRQ Enable for UART[3] (uart3_en):** This bit enables wake from S0ix and generation of De/AssertIRQn message based on UART's PME & D0i3 signals. The resulting De/AssertIRQn will have vector. as programmed in the corresponding GCR.IRQ_SEL_0/1 field.
2	0h RW	**Direct IRQ Enable for UART[2] (uart2_en):** This bit enables wake from S0ix and generation of De/AssertIRQn message based on UART's PME & D0i3 signals. The resulting De/AssertIRQn will have vector. as programmed in the corresponding GCR.IRQ_SEL_0/1 field.
1	0h RW	**Direct IRQ Enable for UART[1] (uart1_en):** This bit enables wake from S0ix and generation of De/AssertIRQn message based on UART's PME & D0i3 signals. The resulting De/AssertIRQn will have vector. as programmed in the corresponding GCR.IRQ_SEL_0/1 field.
0	0h RW	**Direct IRQ Enable for UART[0] (uart0_en):** This bit enables wake from S0ix and generation of De/AssertIRQn message based on UART's PME & D0i3 signals. The resulting De/AssertIRQn will have vector. as programmed in the corresponding GCR.IRQ_SEL_0/1 field.

10.82 Power and Reset Status (PRSTS)—Offset 1000h

Bits in this register only need to be valid for reading when the Main power well is up. However, since some of the events may initially be detected while the Main power well is down, they are marked as suspend well bits. All suspend well bits in this register are reset by global reset#.

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (rsvd_pmc_prodid):** PMC Product ID. Reserved: This field communicates the Product Family of the power management functionality.

Bit Range	Default & Access	Field Name (ID): Description
23:16	0h RO	**Reserved (rsvd_pmc_revid):** PMC Revision ID. Power Management Controller Revision ID. Reference only: This field communicates the implementation revision of the power management functionality.
15	0h RO	**Reserved (rsvd_pmc_wdt_sts):** PMC Watch Dog Timer Status. Reserved: This bit will be set to '1' when the PMC Watch Dog Timer triggers a reset. It will be cleared by a write of '1' by software.
14:8	0h RO	**Reserved (reserved2):** Reserved.
7	0h RW/1C/V	**CSE Clear Reset Status (cse_cldrst_sts):** This bit will be set to '1' when the CSE FW triggers a reset. It will be cleared by a write of '1' by software. BIOS should read GEN_PMCON1 to determine what type of reset CSE requested. Reset by pwrgood.
6	0h RO	**Reserved (rsvd_cse_wdt_sts):** CSE Watch Dog Timer Status. Reserved: This bit will be set to '1' when the CSE Watch Dog Timer triggers a reset. It will be cleared by a write of '1' by software.
5:0	0h RO	**Reserved (reserved1):** Reserved.

10.83 PM CFG - Power Management Configuration (PMC_CFG)—Offset 1008h

This register contains misc. fields used to configure the SOC's power management behavior.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RO	**Reserved (reserved4):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
19:18	0h RO	**Reserved (rsvd_slp_sus_min_asst_wdth):** SLP_SUS# Minimum Assertion Width. Reference only - This 2-bit value indicates the minimum assertion width of the SLP_SUS# signal to guarantee that the SUS Well power supplies have been fully power-cycled. This value may be modified per platform depending on power supply capacitance, board capacitance, power failure detection circuits, etc. Settings are: 2'b00: 0 ms (i.e. stretching disabled - default) 2'b01: 500 ms 2'b10: 1 s 2'b11: 4 s This field is not writable when the SLP_Sx# Stretching Policy Lock Down bit is set. This field is ignored when exiting a G3 state if the Disable SLP_X Stretching After SUS Power Failure bit is set. Note that unlike with all other SLP_* pin stretching, this disable bit only impacts SLP_SUS# stretching during G3 exit rather than both G3 and DeepSx exit. SLP_SUS# stretching always applies to DeepSx regardless of the disable bit. Programming Note: For platforms that enable DeepSx, BIOS must program SLP_SUS# stretching to be greater than or equal to the largest stretching value on any other SLP_* pin (SLP_S3#, SLP_S4#, SLP_LAN#, or SLP_A#). This bit is only fully cleared by the RTEST# pin.
17:16	0h RO	**Reserved (rsvd_slp_sus_a_asst_wdth):** SLP_A# Minimum Assertion Width. Reference only - This 2-bit value indicates the minimum assertion width of the SLP_A# signal to guarantee that the ASW power supplies have been fully power-cycled. This value may be modified per platform depending on power supply capacitance, board capacitance, power failure detection circuits, etc. Settings are: 2'b00: 0 ms (i.e. stretching disabled - default) 2'b01: 4 s 2'b10: 98 ms 2'b11: 2 s This field is not writable when the SLP_Sx# Stretching Policy Lock Down bit is set. This field is ignored when exiting a G3 state if the Disable SLP_X Stretching After SUS Power Failure bit is set.
15:10	0h RO	**Reserved (reserved3):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW/V/L	**Reset Power Cycle Duration (pwr_cyc_dur):** The value in this register determines the minimum time a platform will stay in reset (SLP_S3#, SLP_S4#, SLP_S5# asserted and also SLP_A# and SLP_LAN# asserted if applicable) during a host partition reset (if applicable) with power cycle or a global reset. The duration programmed in this register takes precedence over the applicable SLP_# stretch timers in these reset scenarios. Valid values are: 2'b11: 1-2 seconds 2'b10: 2-3 seconds 2'b01: 3-4 seconds 2'b00: 4-5 seconds (default) This field is not writable when the SLP_Sx# Stretching Policy Lock Down bit is set. Note that the duration programmed in this register should never be smaller than the stretch duration programmed in the following registers: GEN_PMCON_3.SLP_S3_MIN_ASST_WDTH GEN_PMCON_3.S4MAW PM_CFG.SLP_A_MIN_ASST_WDTH (if applicable) PM_CFG_SLP_LAN_MIN_ASST_WDTH (if applicable) These bits are fully cleared by RTCRST# assertion
7:6	0h RO	**Reserved (reserved2):** Reserved.
5	0h RO	**Reserved (rsvd_sps):** Shutdown Policy Select. ITSS is the recipient of CF9 I/O writes. Reference only: When cleared (default) the SOC will drive INIT# in response to the shutdown Message. When set to 1, SOC will treat the shutdown message similar to receiving a CF9h I/O write, and will drive PMU_PLTRST active. This register is reset any time HOST_RST asserts.
4	0h RW	**No Reboot (no_reboot):** FW will set this bit when TCO_NO_REBOOT SMIP is set.SW might also set this bit.When set, the TCO timer will count down and generate the SMI# on the first timeout, but will not reboot on the second timeout.
3	0h RW	**S1/3/4/5 Entry Timeout Enable (sx_ent_to_en):** This policy bit determines whether the SOC will apply a timeout to the S3/S4/S5 entry flow. If this timeout is enabled and the entry flow appears to be hung, the SOC will trigger a global reset. Encodings: 0: Timeout Disabled (default) 1: Timeout Enabled Note: The active SMIP value to continue will override timeout bit for S4/S5 only. Note: BXT0 and BXT1 only support S4 and S5 (as cold off states).
2:0	0h RO	**Reserved (reserved1):** Reserved.

10.84 Power Management Configuration (PMC_CFG2)—Offset 100Ch

This register contains misc. fields used to configure the SOC's power management behavior.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RW	**Power Button Override Period (pwrbtn_ovr_per):** This field determines, while the power button remains asserted, how long the PMC will wait before initiating a global reset. Encoding: 3'b000: 4 seconds 3'b001: 6 seconds 3'b010: 8 seconds 3'b011: 10 seconds 3'b100: 12 seconds 3'b101: 14 seconds Others: Reserved
28	0h RW/V/L	**Power Button Native Mode Disable (pwrbtn_dis):** When this bit is '0' (default), the PMC's power button logic will act upon the input value from the GPIO unit, as normal. When this bit is set to '1', the PMC must force its internal version of the power button pin to '1'. This will result in the PMC logic constantly seeing the pin as de-asserted.
27:11	0h RO	**Reserved (reserved0):** Reserved.
10	0h RW/V	**Power Button Debounce Mode (pwrbtn_db_mode):** This bit controls when interrupts (SMI#, SCI) are generated in response to assertion of the PWRBTN# pin. This bit's values cause the following behavior: 0: The 16 ms debounce period applies to all usages of the PWRBTN# pin (legacy behavior). 1: When a falling edge occurs on the PWRBTN# pin, an interrupt is generated and the 16 ms debounce timer starts. Subsequent interrupts are masked while the debounce timer is running. Note: Power button override logic always samples the post-debounce version of the pin.
9:0	0h RO	**Reserved (reserved1):** Reserved.

10.85 SOC Power Management Status (SOC_PM_STS)—Offset 1010h

This register contains misc. fields used to record events pertaining to SOC power management. Unless otherwise indicated, all RWC bits are cleared with a write of 1 by software.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RO	**Reserved (rsvd):** Reserved.
2	0h RO	**Reserved (rsvd_hpr_ent_to):** Host Partition Reset Entry Timeout. This bit is set to '1' to record that a global reset was triggered by a timeout during Host partition reset entry sequence.
1	0h RO	**Reserved (rsvd_sx_ent_to):** S3/4/5 Entry Timeout. Reset cause information is available via Shared SRAM Reserved: This bit is set to '1' to record that a global reset was triggered by a timeout during an S3, S4, or S5 entry sequence.
0	0h RO	**Reserved (reserved1):** Reserved.

10.86 General PM Configuration 1 (GEN_PMCON1)—Offset 1020h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 4004h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (reserved6):** Reserved.
27	0h RW/1C/V	**Cold Boot Status (cold_boot_sts):** FW will populate this BIOS visible bit based on a COLD BOOT. This bit acts as notification to BIOS of the type of boot/reset.
26	0h RW/1C/V	**Cold Reset Status (cold_reset_sts):** FW will populate this BIOS visible bit based on a COLD RESET. This bit acts as notification to BIOS of the type of boot/reset.
25	0h RW/1C/V	**Warm Reset Status (warm_reset_sts):** FW will populate this BIOS visible bit based on a WARM RESET. This bit acts as notification to BIOS of the type of boot/reset.

Bit Range	Default & Access	Field Name (ID): Description
24	0h RW/1C/V	**Global Reset Status (global_reset_sts):** FW will populate this BIOS visible bit based on a GLOBAL RESET. This bit acts as notification to BIOS of the type of boot/reset.
23	0h RW	**DRAM Initialization Scratchpad Bit (disb):** This bit does not effect hardware functionality in any way. It is provided as a scratchpad bit that is maintained through main power well resets and CF9h-initiated resets. BIOS is expected to set this bit prior to starting the DRAM initialization sequence and to clear this bit after completing the DRAM initialization sequence. BIOS can detect that a DRAM initialization sequence was interrupted by a reset by reading this bit during the boot sequence. If the bit is 1, then the DRAM initialization was interrupted.
22	0h RO	**Reserved (reserved5):** Reserved.
21	0h RO	**Reserved (rsvd_mem_sr):** Memory Placed in Self-Refresh. Reference only: This bit will be set to 1 if DRAM should have remained powered and held in Self-Refresh through the last power state transition (i.e. the last time the system left S0). The scenarios where this should be the case are: Successful S3 entry & exit Successful Host partition reset without power cycle This bit will be cleared whenever the SOC begins a transition out of S0. Note: This bit should not be consulted upon wake from S1, as that state does not involve the same type of handshake or placing memory into Self-Refresh. It is assumed that software is already aware that memory context is not impacted by S1 and therefore does not need to check this bit.
20	0h RW/1C/V	**System Reset Status (srs):** This bit will be set based on an asserting edge of the PMU_RESETBUTTON_B pin (post 16 ms HW debounce if PMU_RESETBUTTON_B Debounce Disable SMIP is 0) BIOS is expected to read this bit and clear it if it is set.
19	0h RO	**Reserved (rsvd_cts):** CPU Thermal Trip Status. Reference only: This bit is set when the SOC thermal trip active while the system is in a valid state to honor the pin. This bit is also reset by RSMRST_B and CF9h resets. It is not reset by the shutdown and reboot associated with the thermal trip event. RSMRST_B is a reset signal based on a RSMRST# pin present on some PCH platforms that indicates the suspend/resume voltages are stable. Note: This bit is must for no PMIC platform. In platforms with PMIC - PMIC handles the thermal trip. Hence, actual Thermal Trip Status bit should be part of PMIC registers, while PMC could mirror it to this bit.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW/1C/V	**Minimum PMU SLP S4 B Assertion Width Violation Status (ms4v):** Hardware sets this bit when the PMU_SLP_S4_B assertion width is less than the time programmed in the PMU_SLP_S4_B Minimum Assertion Width field. The SOC begins the timer when PMU_SLP_S4_B pin is asserted during S4/S5 entry, or when the RSMRST_B input is deasserted during SUS well power-up. The status bit is cleared by software writing a 1 to the bit. Note that this bit is functional regardless of the value in the PMU_SLP_S4_B Assertion Stretch Enable and the Disable-SLP_X-Stretching-After-SUS-Power-Failure bits. This bit is reset by the assertion of the RSMRST_B pin, but can be set in some cases before the default value is readable. Note: Relevant only for no PMIC mode.
17	0h RO	**Reserved (reserved4):** Reserved
16	0h RO	**Reserved (rsvd_pwr_flr):** VNNPWROK Failure. Reference only: Intel SOC sets this bit any time COREPWROK goes low if the system was in an S0 state. The bit will be cleared only by software writing a 1 back to the bit or by SUS well power loss. Note: This bit is must for no PMIC platform. In platforms with PMIC - PMIC handles Vnn failures. Hence, actual VNNPWROK Failure bit should be part of PMIC registers, while PMC could mirror it to this bit.
15	0h RW/V/L	**PME B0 S5 Disable (pme_b0_s5_dis):** When set to '1', this bit blocks wake events from PME_B0_STS in S5, regardless of the state of PME_B0_EN. When cleared (default), wake events from PME_B0_STS are allowed in S5 if PME_B0_EN = '1'. Wakes from power states other than S5 are not affected by this policy bit. It has a separate STS and EN for each source. Therefore this bit affects GPE0a_STS/EN PME bits for: XDCI, XHCI, CNV, and AVS.
14	1h RO	**Reserved (rsvd_aon_pwr_flr):** VNNAON Well Power Failure. This bit is set to '1' whenever SUS well power is lost, as indicated by RSMRST_B assertion. Software writes a 1 to this bit to clear it. This bit is in the SUS well, and defaults to '1' based on RSMRST_B assertion (not cleared by any type of reset). Note: This bit is must for no PMIC platform. In platforms with PMIC - PMIC handles VnnAON failures. Hence, actual VNNAON Well Power Failure bit should be part of PMIC registers, while PMC could mirror it to this bit.
13:10	0h RO	**Reserved (Reserved3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9	0h RO	**Reserved (rsvd_gen_rst_sts):** General Reset Status. Reference only: This bit is set by hardware whenever HOST_RST asserts for any reason other than going into a software-entered sleep state (via PM1_CNT.SLP_EN write). This bit is an optional tool to help BIOS determine when a reset might have collided with a wake from a valid sleep state. A possible usage model would be to consult and then write a '1' to clear this bit during the boot flow before determining what action to take based on reading PM1_STS.WAK_STS = '1'. If GEN_RST_STS = '1', the cold reset boot path could be followed rather than the resume path, regardless of the setting of WAK_STS. This bit does not affect SOC operation in any way, and can therefore be left set if BIOS chooses not to use it. This bit is set by global reset.
8	0h RO	**Reserved (Reserved2):** Reserved.
7:6	0h RW	**SWSMI Rate Select (swsmi_ratesel):** This 2-bit value indicates when the SWSMI timer will time out. Valid values are: 00 1.5 ms +/- 0.6 ms 01 16 ms +/- 4 ms 10 32 ms +/- 4 ms 11 64 ms +/- 4 ms
5:3	0h RO	**Reserved (reserved1):** Reserved.
2	1h RW/V	**RTC Power Status (rps):** Intel SOC will set this bit to 1 when RTEST_B indicates a weak or missing battery. The bit will remain set until the software clears it by writing a 0 back to this bit position. This bit is not cleared by any type of reset because a cleared shadowed value will be restored prior to this bit being visible. Note: Relevant only for no PMIC mode while SoC includes RTC well.
1	0h RO	**Reserved (reserved0):** Reserved.
0	0h RW/V/L	**After G3 Enable (ag3e):** Determines what state to go to when power is reapplied after a power failure (G3 state). 0: System will return to an S0 state (boot) after power is re-applied. 1: System will return to the S5 state (except if it was in S4, in which case it will return to S4-like state). In the S5 state, the only enabled wake-up event is the Power Button or any enabled wake event that was preserved through the power failure. Note: Relevant only for no PMIC mode.

10.87 General PM Configuration 2 (GEN_PMCON2)—Offset 1024h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 3800h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**Reserved (reserved1):** Reserved.
13:11	7h RW/L	**LPC Loopback Clock Control (lpc_lpb_clk_ctrl):** Delay for LPC loopback clock (to match board latency to and from the embedded controller. 000: RESERVED (This setting is not supported and may lead to inconsistent behavior on silicon) 001: 1.25 ns 010: 3.75 ns 011: 6.25 ns 100: 8.75 ns 101: 11.25 ns 110: 12.50 ns 111: 13.75 ns
10	0h RO	**Reserved (rsvd_bios_pci_exp_en):** BIOS PCI Express Enable. Reference only: This bit acts as a global enable for the SCI associated with the PCI express ports. If this bit is not set, then the various PCI Express ports cannot cause the PCI_EXP_STS bit to go active.
9	0h RO/V	**Power Button Level (pwrbtn_lvl):** This read-only bit indicates the current state of the PWRBTN# signal. 1: High 0: Low The value reflected in this bit is dependent upon PMC_CFG2.pwrbtn_db_mode
8:5	0h RO	**Reserved (reserved3):** Reserved.
4	0h RW/L	**SMI Lock (smi_lock):** When this bit is set, writes to the GBL_SMI_EN bit will have no effect. Once the SMI_LOCK bit is set, writes of 0 to SMI_LOCK bit will have no effect (i.e. once set, this bit can only be cleared by reset).
3:2	0h RO	**Reserved (reserved4):** Reserved.
1:0	0h RW/L	**Period SMI Select (per_smi_sel):** Software sets these bits to control the rate at which the periodic SMI# is generated: 00: 64 seconds (default) 01 = 32 seconds 10 = 16 seconds 11 = 8 seconds Tolerance for the timer is +/- 1 second.

10.88 General PM Configuration 3 (GEN_PMCON3)—Offset 1028h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:13	0h RO	**Reserved (reserved1):** Reserved.
12	0h RO	**Reserved (rsvd_dis_slp_x_strch_sus_up):** Disable SLP_X Stretching After SUS Well Power Up. Reference only: When this bit is set to 1, all SLP_* pin stretching is disabled when powering up after a SUS well power loss. When this bit is left at 0, SLP_* stretching will be performed after SUS power failure as enabled in various other fields. Note that if this bit is a 0, SLP_* stretch timers start on SUS well power up (the SOC has no ability to count stretch time while the SUS well is powered down). Setting this bit can therefore prevent long delays after SUS power loss which may be common in mobile platforms and in manufacturing flow testing, while still allowing for the full power cycling during S3, S4 and S5 states. If the platform guarantees minimum SUS power down residence in other ways, an additional SOC-induced delay is not needed or wanted. This field is not writable when the SLP_Sx# Stretching Policy Lock-Down bit is set. This bit is cleared by the SRTCRST_B pin. Note: Relevant only for no PMIC mode.
11:10	0h RW/L	**PMU SLP S3 B Minimum Assertion Width (slp_s3_min_asst_wdth):** Reference only: This 2-bit value indicates the minimum assertion width of the PMU_SLP_S3_B signal to guarantee that the Main power supplies have been fully power-cycled. This value may be modified per platform depending on power supply capacitance, board capacitance, power failure detection circuits, etc. Settings are: 00: 60 usec 01: 1 ms 10: 50 ms 11: 2 sec This field is not writable when the SLP_Sx# Stretching Policy Lock-Down bit is set. Note: Relevant only for no PMIC mode.
9:6	0h RO	**Reserved (reserved2):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
5:4	0h RW/V/L	**PMU SLP S4 B Minimum Assertion Width (s4maw):** This 2-bit value indicates the minimum assertion width of the PMU_SLP_S4_B signal to guarantee that the DRAMs have been safely power-cycled. This value may be modified per platform depending on DRAM types, power supply capacitance, etc. Valid values are: 11: 1 second 10: 2 seconds 01: 3 seconds 00: 4 seconds This value is used in two ways: If the PMU_SLP_S4_B assertion width is ever shorter than this time, a status bit is set for BIOS to read when S0 is entered. If enabled by bit 3 in this register, the hardware will prevent the PMU_SLP_S4_B signal from deasserting within this minimum time period after asserting. Note that the logic that measures this time is in the suspend power well. Therefore, when leaving a G3 state, the minimum time is measured from the deassertion of the internal suspend well reset (unless the Disable SLP_X Stretching After SUS Power Failure bit is set). This field is not writable when the SLP_Sx# Stretching Policy Lock-Down bit is set. SRTCRST_B forces this field to the conservative default state (00b). Note: Relevant only for no PMIC mode.
3	0h RW/V/L	**PMU SLP S4 B Assertion Stretch Enable (s4ase):** When set to 1, the PMU_SLP_S4_B pin will minimally assert for the time specified in bits 5:4 of this register. When 0, the reset flow still includes the SLP_S3_MIN_ASST_WDTH but no additional time is included for S4ASE. This bit is provided so that all DIMMs in the system can deterministically detect a power-cycle event for proper initialization. Note that there are behavioral changes that may be noticeable to the end-user when this bit is set. Resume times from S4 and S5 and power-up times from G3 may be delayed by several seconds. Cases in which this feature may not be desirable and therefore keeping the bit cleared are: A customer decides the user confusion due to the hardware delay is a bigger issue than the potential DRAM issue A customer decides the software status bit solution is adequate A different DRAM type is used or the platform provides an external solution to solve the power-cycling issue Validation regressions are impacted by the delay (especially after RSMRST_B deassertion) Avoid potential resume time WHQL violations This bit is not writable when the SLP_Sx# Stretching Policy Lock-Down bit is set. This bit is cleared by SRTCRST_B. Note: Relevant only for no PMIC mode.
2:0	0h RO	**Reserved (reserved3):** Reserved.

10.89 Configured Revision ID (CRID)—Offset 1030h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Reserved (rsvd):** Reserved.
1:0	0h RW/L	**RID Select (rid_sel):** Software writes this field to select the fuse sets reflected in PCI config space Revision ID. The decoding is: 00: Revision ID 01: CRID 0 10: CRID 1 11: CRID 2 (setid multicast sends this out)

10.90 Function Disable 0 (FUNC_DIS_0)—Offset 1034h

BIOS uses this register to disable specific function. PMC FW consults this register during reset to determine whether to bring an IP out of reset (applicable to Audio and ISH only for BXT0).

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/L	**GMM Disable (gmm):** Set by BIOS to inform PMC GMM is disabled.
30	0h RW/L	**SPI Disable (spi):** Set by BIOS to inform PMC SPI is disabled.
29	0h RW/L	**SSRAM Disable (ssram):** Set by BIOS to inform PMC SSRAM is disabled.
28	0h RW/L	**AVS Disable (avs):** Set by BIOS to inform PMC AVS is disabled.
27	0h RW/L	**CSE HECI1 Disable (cse_heci1):** Set by BIOS to inform PMC CSE_HECI1 is disabled. CSE_HECI1 is function 0 of a Multi-function device. If this function is disabled all functions of this device must be disabled.
26	0h RW/L	**CSE HECI2 Disable (cse_heci2):** Set by BIOS to inform PMC CSE_HECI2 is disabled.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW/L	**CSE HECI3 Disable (cse_heci3):** Set by BIOS to inform PMC CSE_HECI3 is disabled.
24	0h RW/L	**ISH Disable (ish):** Set by BIOS to inform PMC ISH is disabled.
23	0h RW/L	**PCIE0P0 Disable (pcie0p0):** Set by BIOS to inform PMC PCIE0 port 0 is disabled.
22	0h RW/L	**PCIE0P1 Disable (pcie0p1):** Set by BIOS to inform PMC PCIE0 port 1 is disabled.
21	0h RW/L	**LPSS I2C0 Disable (lpss_i2c0):** Set by BIOS to inform PMC LPSS_I2C0 is disabled. LPSS_I2C0 is function 0 of a Multi-function device. If this function is disabled all functions of this device must be disabled.
20	0h RW/L	**LPSS I2C1 Disable (lpss_i2c1):** Set by BIOS to inform PMC LPSS_I2C1 is disabled.
19	0h RW/L	**LPSS I2C2 Disable (lpss_i2c2):** Set by BIOS to inform PMC LPSS_I2C2 is disabled.
18	0h RW/L	**LPSS I2C3 Disable (lpss_i2c3):** Set by BIOS to inform PMC LPSS_I2C3 is disabled.
17	0h RW/L	**LPSS I2C4 Disable (lpss_i2c4):** Set by BIOS to inform PMC LPSS_I2C4 is disabled. LPSS_I2C4 is function 0 of a Multi-function device. If this function is disabled all functions of this device must be disabled.
16	0h RW/L	**LPSS I2C5 Disable (lpss_i2c5):** Set by BIOS to inform PMC LPSS_I2C5 is disabled.
15	0h RW/L	**LPSS I2C6 Disable (lpss_i2c6):** Set by BIOS to inform PMC LPSS_I2C6 is disabled.
14	0h RW/L	**LPSS I2C7 Disable (lpss_i2c7):** Set by BIOS to inform PMC LPSS_I2C7 is disabled.
13	0h RW/L	**LPSS UART0 Disable (lpss_uart0):** Set by BIOS to inform PMC LPSS_UART0 is disabled. LPSS_UART0 is function 0 of a Multi-function device. If this function is disabled all functions of this device must be disabled.
12	0h RW/L	**LPSS UART1 Disable (lpss_uart1):** Set by BIOS to inform PMC LPSS_UART1 is disabled.
11	0h RW/L	**LPSS UART2 Disable (lpss_uart2):** Set by BIOS to inform PMC LPSS_UART2 is disabled.
10	0h RW/L	**LPSS UART3 Disable (lpss_uart3):** Set by BIOS to inform PMC LPSS_UART3 is disabled.
9	0h RW/L	**LPSS SPI0 Disable (lpss_spi0):** Set by BIOS to inform PMC LPSS_SPI0 is disabled. LPSS_SPI0 is function 0 of a Multi-function device. If this function is disabled all functions of this device must be disabled.
8	0h RW/L	**LPSS SPI1 Disable (lpss_spi1):** Set by BIOS to inform PMC LPSS_SPI1 is disabled.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW/L	**LPSS SPI2 Disable (lpss_spi2):** Set by BIOS to inform PMC LPSS_SPI2 is disabled.
6	0h RW/L	**PWM Disable (pwm):** Set by BIOS to inform PMC PWM is disabled.
5	0h RW/L	**SDCARD Disable (sdcard):** Set by BIOS to inform PMC SDCARD is disabled.
4	0h RW/L	**SDIO Disable (sdio):** Set by BIOS to inform PMC SDIO is disabled.
3	0h RW/L	**eMMC Disable (emmc):** Set by BIOS to inform PMC eMMC is disabled.
2	0h RW/L	**UFS Disable (ufs):** Set by BIOS to inform PMC UFS is disabled.
1	0h RW/L	**xHCI Disable (xhci):** Set by BIOS to inform PMC xHCI is disabled. xHCI is function 0 of a Multi-function device. If this function is disabled all functions of this device must be disabled.
0	0h RW/L	**xDCI Disable (xdci):** Set by BIOS to inform PMC xDCI is disabled.

10.91 Function Disable 1 (FUNC_DIS_1)—Offset 1038h

BIOS uses this register to disable specific function. PMC FW consults this register during reset to determine whether to bring an IP out of reset (applicable to Audio and ISH only for BXT0).

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RO	**Reserved (rsvd31):** Reserved.
9	0h RW/L	**SATA Disable (sata):** Set by BIOS to inform PMC SATA is disabled.
8:7	0h RO	**Reserved PCIE Disable (rsvd_pcie):** Reserved for PCIe expansion.
6	0h RW/L	**PCIE1 Port 3 Disable (pcie1p3):** Set by BIOS to inform PMC PCIE1 port 3 is disabled.
5	0h RW/L	**PCIE1 Port 2 Disable (pcie1p2):** Set by BIOS to inform PMC PCIE1 port 2 is disabled.
4	0h RW/L	**PCIE1 Port 1 Disable (pcie1p1):** Set by BIOS to inform PMC PCIE1 port 1 is disabled.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW/L	**PCIE1 Port 0 Disable (pcie1p0):** Set by BIOS to inform PMC PCIE1 port 0 is disabled.
2	0h RW/L	**UFS Disable (ufs2):** Set by BIOS to inform PMC that 3rd lane UFS is disabled.
1	0h RW/L	**CNV Disable (cnv):** Set by BIOS to inform PMC that CNV is disabled.
0	0h RO	**Reserved (rsvd0):** Reserved.

10.92 Extended Test Mode Register (ETR)—Offset 1048h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 38000h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/L	**CF9h Lockdown (cf9lock):** When set, this will lock the CF9h-Global-Reset bit. When set, this register locks itself.
30:21	0h RO	**Reserved (reserved3):** Reserved.
20	0h RW/L	**CF9h Global Reset (cf9gr):** When this bit is set, a CF9h write of 6h or Eh will cause a Global Reset (of the Host partition where the designation is applicable). If this bit is cleared, a CF9h write of 6h or Eh will only reset the Host partition (Cold or Warm reset were no Host partition.) It is recommended that BIOS should set this bit early on in the boot sequence, and then clear it and set the CF9LOCK bit prior to loading the OS. When this bit is set, the hardware assumes that bit 18 (CF9h Without Resume Well Reset Enable) is cleared. This register is locked by the CF9 Lockdown (CF9LOCK) bit.
19:18	0h RO	**Reserved (reserved2):** Reserved.
17:15	7h RW/L	**Max S0IX (max_s0ix):** Indicated the maximum S0i state SOC can enter. 111: LTR/TNTE L6 threshold (S0i3) - Deepest S0ix state allowed 110: LTR/TNTE L6 threshold (S0i3) 101: LTR/TNTE L5 threshold 100: LTR/TNTE L4 threshold 011: LTR/TNTE L3 threshold 010: LTR/TNTE L2 threshold (Fastest Exit/Highest Pwr) 001: LTR/TNTE L1 threshold 000: No S0ix state allowed
14:0	0h RO	**Reserved (reserved1):** Reserved.

10.93 GPIO Group to General Purpose Event Register Configuration (GPIO_GPE_CFG)—Offset 1050h

Maps the ACPI GPE0[b-d] registers to GPIO groups. Each of gpe0_dw[1-3] is a lookup which maps a read/write to gpe0[b-d]_[en,sts] to a particular GPIO community and offset.

General purpose event enable register address mapping
15: 16'h150 (reserved)
14: 16'h150 (reserved)
13: 16'h150 (reserved)
12: 16'h150 (reserved)
11: 16'h158
10: 16'h154
09: 16'h150
08: 16'h158
07: 16'h154
06: 16'h150
05: 16'h15C
04: 16'h158
03: 16'h154
02: 16'h150
01: 16'h154
00: 16'h150

General purpose event status register address mapping
15: 16'h130 (reserved)
14: 16'h130 (reserved)
13: 16'h130 (reserved)
12: 16'h130 (reserved)
11: 16'h138
10: 16'h134
09: 16'h130
08: 16'h138
07: 16'h134
06: 16'h130
05: 16'h13C
04: 16'h138
03: 16'h134
02: 16'h130
01: 16'h134
00: 16'h130

GPIO Community PortID mapping
15: SB_PORTID_GPIOSCC (reserved)
14: SB_PORTID_GPIOSCC (reserved)
13: SB_PORTID_GPIOSCC (reserved)
12: SB_PORTID_GPIOSCC (reserved)
11: SB_PORTID_GPIOAUD
10: SB_PORTID_GPIOAUD
09: SB_PORTID_GPIOAUD
08: SB_PORTID_GPN_CORE
07: SB_PORTID_GPN_CORE
06: SB_PORTID_GPN_CORE

05: SB_PORTID_GPNW_CORE
04: SB_PORTID_GPNW_CORE
03: SB_PORTID_GPNW_CORE
02: SB_PORTID_GPNW_CORE
01: SB_PORTID_GPIOSCC
00: SB_PORTID_GPIOSCC

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 6300h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (rsvd31):** Reserved.
15:12	6h RW	**GPIO Group to General Purpose Event Dword 3 (gpe0_dw3):** This register assigns a specific GPIO Group to the ACPI GPE0d (i.e. GPE0[127:96]). The value in this field determines which GPIO group is assigned to dword 3 (gpe0d).
11:8	3h RW	**GPIO Group to General Purpose Event Dword 2 (gpe0_dw2):** This register assigns a specific GPIO Group to the ACPI GPE0c (i.e. GPE0[95:64]). The value in this field determines which GPIO group is assigned to dword 2 (gpe0c).
7:4	0h RW	**GPIO Group to General Purpose Event Dword 1 (gpe0_dw1):** This register assigns a specific GPIO Group to the ACPI GPE0b (i.e. GPE0[63:32]). The value in this field determines which GPIO group is assigned to dword 1 (gpe0b).
3:0	0h RO	**Reserved (rsvd_gpe0_dw0):** GPIO Group to General Purpose Event Dword 0. This register assigns a specific GPIO Group to the ACPI GPE0a (i.e. GPE0[31:0]). The value in this field determines which GPIO group is assigned to dword 0 (gpe0a). This register is reserved / unused, because gpe0a is a local ACPI register.

10.94 UART IRQ Configuration (IRQ_CFG_UART)—Offset 1060h

Configuration for UART Direct IRQ logic to allow inversion or disabling of the CTS or RXD interrupts.
Configured by IAFW to allow for wake support for a variety of UART devices without requiring runtime driver configuration of GPIOs.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: FF00h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (rsvd31):** Reserved.
15	1h RW/L	**UART3 RXD Direct IRQ Enable (uart3_rxd_int_en):** If set, enables RXD wire to cause Direct IRQ.
14	1h RW/L	**UART2 RXD Direct IRQ Enable (uart2_rxd_int_en):** If set, enables RXD wire to cause Direct IRQ.
13	1h RW/L	**UART1 RXD Direct IRQ Enable (uart1_rxd_int_en):** If set, enables RXD wire to cause Direct IRQ.
12	1h RW/L	**UART0 RXD Direct IRQ Enable (uart0_rxd_int_en):** If set, enables RXD wire to cause Direct IRQ.
11	1h RW/L	**UART3 CTS Direct IRQ Enable (uart3_cts_int_en):** If set, enables CTS wire to cause Direct IRQ.
10	1h RW/L	**UART2 CTS Direct IRQ Enable (uart2_cts_int_en):** If set, enables CTS wire to cause Direct IRQ.
9	1h RW/L	**UART1 CTS Direct IRQ Enable (uart1_cts_int_en):** If set, enables CTS wire to cause Direct IRQ.
8	1h RW/L	**UART0 CTS Direct IRQ Enable (uart0_cts_int_en):** If set, enables CTS wire to cause Direct IRQ.
7	0h RW/L	**UART3 RXD invert (uart3_rxd_inv):** If set, inverts RXD wire at the input of the PMC Direct IRQ logic.
6	0h RW/L	**UART2 RXD invert (uart2_rxd_inv):** If set, inverts RXD wire at the input of the PMC Direct IRQ logic.
5	0h RW/L	**UART1 RXD invert (uart1_rxd_inv):** If set, inverts RXD wire at the input of the PMC Direct IRQ logic.
4	0h RW/L	**UART0 RXD invert (uart0_rxd_inv):** If set, inverts RXD wire at the input of the PMC Direct IRQ logic.
3	0h RW/L	**UART3 CTS invert (uart3_cts_inv):** If set, inverts CTS wire at the input of the PMC Direct IRQ logic.
2	0h RW/L	**UART2 CTS invert (uart2_cts_inv):** If set, inverts CTS wire at the input of the PMC Direct IRQ logic.
1	0h RW/L	**UART1 CTS invert (uart1_cts_inv):** If set, inverts CTS wire at the input of the PMC Direct IRQ logic.
0	0h RW/L	**UART0 CTS invert (uart0_cts_inv):** If set, inverts CTS wire at the input of the PMC Direct IRQ logic.

10.95 IRQ Select 0 (IRQ_SEL_0)—Offset 1064h

ACPI uses these fields to fill the IRQ vector of the De/Assert_IRQn message for Direct IRQs.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW/L	**IRQ Vector For UART 3 (dir_irq_sel_uart3)**
23:16	0h RW/L	**IRQ Vector For UART 2 (dir_irq_sel_uart2)**
15:8	0h RW/L	**IRQ Vector For UART 1 (dir_irq_sel_uart1)**
7:0	0h RW/L	**IRQ Vector For UART 0 (dir_irq_sel_uart0)**

10.96 IRQ Select 1 (IRQ_SEL_1)—Offset 1068h

ACPI uses these fields to fill the IRQ vector of the De/Assert_IRQn message for Direct IRQs.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW/L	**IRQ Vector For XHCI (dir_irq_sel_xhci)**
23:16	0h RW/L	**IRQ Vector For XDCI (dir_irq_sel_xdci)**
15:8	0h RW/L	**IRQ Vector For SDCARD CD (dir_irq_sel_sdcard_cd)**
7:0	0h RW/L	**IRQ Vector For SDCARD D1 (dir_irq_sel_sdcard_d1)**

10.97 IRQ Select 2 (IRQ_SEL_2)—Offset 106Ch

ACPI uses these fields to fill the IRQ vector of the De/Assert_IRQn message for Direct IRQs.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW/L	**SCI IRQ Select (scis):** Specifies on which IRQ the SCI will internally appear. ACPI uses this field to fill the IRQ vector of the De/Assert_IRQn message for IOAPIC IRQ for PMIC interrupts. If not using the APIC, the SCI must be routed to IRQ[9-11], and that interrupt is not shareable with the SERIRQ stream, but is shareable with other PCI interrupts. If using the APIC, the SCI can also be mapped to IRQ20-23, and can be shared with other interrupts. This field fully defines the IRQ number to be sent. When the interrupt is mapped to APIC interrupts 9, 10 or 11, the APIC should be programmed for active-high reception. When the interrupt is mapped to APIC interrupts 20 through 23, the APIC should be programmed for active-low reception.
23:16	0h RW/L	**PMIC Direct IRQ Select (dir_irq_sel_pmic):** ACPI uses this field to fill the IRQ vector of the De/Assert_IRQn message for IOAPIC IRQ for PMIC interrupts.
15:0	0h RO	**Reserved (rsvd):** Reserved.

10.98 Function ACPI Enumeration 0 (FUNC_ACPI_ENUM_0)—Offset 1070h

BIOS uses this register to enumerate ACPI functions.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved Enable (rsvd31):** Reserved.
30	0h RW	**SPI Enable (spi):** If set by BIOS the named function is exposed to the OS via ACPI.
29	0h RW	**SSRAM Enable (ssram):** If set by BIOS the named function is exposed to the OS via ACPI.
28	0h RW	**AVS Enable (avs):** If set by BIOS the named function is exposed to the OS via ACPI.
27	0h RW	**CSE HECI1 Enable (cse_heci1):** If set by BIOS the named function is exposed to the OS via ACPI.
26	0h RW	**CSE HECI2 Enable (cse_heci2):** If set by BIOS the named function is exposed to the OS via ACPI.
25	0h RW	**CSE HECI3 Enable (cse_heci3):** If set by BIOS the named function is exposed to the OS via ACPI.

Bit Range	Default & Access	Field Name (ID): Description
24	0h RW	**ISH Enable (ish):** If set by BIOS the named function is exposed to the OS via ACPI.
23	0h RO	**Reserved (rsvd23):** Reserved.
22	0h RO	**Reserved (rsvd22):** Reserved.
21	0h RW	**LPSS I2C0 Enable (lpss_i2c0):** If set by BIOS the named function is exposed to the OS via ACPI.
20	0h RW	**LPSS I2C1 Enable (lpss_i2c1):** If set by BIOS the named function is exposed to the OS via ACPI.
19	0h RW	**LPSS I2C2 Enable (lpss_i2c2):** If set by BIOS the named function is exposed to the OS via ACPI.
18	0h RW	**LPSS I2C3 Enable (lpss_i2c3):** If set by BIOS the named function is exposed to the OS via ACPI.
17	0h RW	**LPSS I2C4 Enable (lpss_i2c4):** If set by BIOS the named function is exposed to the OS via ACPI.
16	0h RW	**LPSS I2C5 Enable (lpss_i2c5):** If set by BIOS the named function is exposed to the OS via ACPI.
15	0h RW	**LPSS I2C6 Enable (lpss_i2c6):** If set by BIOS the named function is exposed to the OS via ACPI.
14	0h RW	**LPSS I2C7 Enable (lpss_i2c7):** If set by BIOS the named function is exposed to the OS via ACPI.
13	0h RW	**LPSS UART0 Enable (lpss_uart0):** If set by BIOS the named function is exposed to the OS via ACPI.
12	0h RW	**LPSS UART1 Enable (lpss_uart1):** If set by BIOS the named function is exposed to the OS via ACPI.
11	0h RW	**LPSS UART2 Enable (lpss_uart2):** If set by BIOS the named function is exposed to the OS via ACPI.
10	0h RW	**LPSS UART3 Enable (lpss_uart3):** If set by BIOS the named function is exposed to the OS via ACPI.
9	0h RW	**LPSS SPI0 Enable (lpss_spi0):** If set by BIOS the named function is exposed to the OS via ACPI.
8	0h RW	**LPSS SPI1 Enable (lpss_spi1):** If set by BIOS the named function is exposed to the OS via ACPI.
7	0h RW	**LPSS SPI2 Enable (lpss_spi2):** If set by BIOS the named function is exposed to the OS via ACPI.
6	0h RW	**PWM Enable (pwm):** If set by BIOS the named function is exposed to the OS via ACPI.
5	0h RW	**SDCARD Enable (sdcard):** If set by BIOS the named function is exposed to the OS via ACPI.
4	0h RW	**SDIO Enable (sdio):** If set by BIOS the named function is exposed to the OS via ACPI.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**eMMC Enable (emmc):** If set by BIOS the named function is exposed to the OS via ACPI.
2	0h RW	**UFS Enable (ufs):** If set by BIOS the named function is exposed to the OS via ACPI.
1	0h RO	**Reserved (rsvd1):** Reserved.
0	0h RW	**xDCI Enable (xdci):** If set by BIOS the named function is exposed to the OS via ACPI.

10.99 Function ACPI Enumeration 1 (FUNC_ACPI_ENUM_1)—Offset 1074h

BIOS uses this register to enumerate ACPI functions.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RO	**Reserved (rsvd31):** Reserved.
2	0h RW	**USF2 Enable (ufs2):** If set by BIOS the named function (3rd lane UFS) is exposed to the OS via ACPI
1	0h RW	**PMC PCI Enable (pmc_pci):** If set by BIOS the named function is exposed to the OS via ACPI.
0	0h RW	**P2SB Enable (p2sb):** If set by BIOS the named function is exposed to the OS via ACPI.

10.100 Fixed Deep S0Ix Counter Lower 32 Bits (TELEM_DEEP_S0IX_LO_HOST)—Offset 1078h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Fixed Deep S0Ix Counter (telem_deep_s0ix_lo_host)**

10.101 Fixed Deep S0Ix Counter Upper 32 Bits (TELEM_DEEP_S0IX_HI_HOST)—Offset 107Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Fixed Deep S0Ix Counter Upper 32 Bits (telem_deep_s0ix_hi_host)**

10.102 Fixed Shallow S0Ix Counter Lower 32 Bits (TELEM_SHALLOW_S0IX_LO_HOST)—Offset 1080h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Fixed Shallow S0Ix Counter (telem_shallow_s0ix_lo_host)**

10.103 Fixed Shallow S0Ix Counter Upper 32 Bits (TELEM_SHALLOW_S0IX_HI_HOST)—Offset 1084h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Fixed Shallow S0Ix Counter (telem_shallow_s0ix_hi_host)**

10.104 Reserved (TELEM_MISC_FIXED_LO_HOST)—Offset 1088h

Register for sharing data with software lower 32 bits.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Reserved (telem_misc_fixed_lo_host):** Reserved register for sharing data with software.

10.105 Reserved register for sharing data with software upper 32 bits (TELEM_MISC_FIXED_HI_HOST)—Offset 108Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Reserved (telem_misc_fixed_hi_host):** Reserved register for sharing data with software.

10.106 BIOS scratchpad (BIOS_SCRATCHPAD)—Offset 1090h

Scratchpad for sharing data between BIOS and PMC Firmware

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 8h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/V	**Scratchpad 31 (scratchpad31):** Scratchpad bit
30	0h RW/V	**Scratchpad 30 (scratchpad30):** Scratchpad bit
29	0h RW/V	**Scratchpad 29 (scratchpad29):** Scratchpad bit
28	0h RW/V	**Scratchpad 28 (scratchpad28):** Scratchpad bit
27	0h RW	**Scratchpad 27 (scratchpad27):** Scratchpad bit
26	0h RW	**Scratchpad 26 (scratchpad26):** Scratchpad bit

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**Scratchpad 25 (scratchpad25):** Scratchpad bit
24	0h RW	**Scratchpad 24 (scratchpad24):** Scratchpad bit
23	0h RW	**Scratchpad 23 (scratchpad23):** Scratchpad bit
22	0h RW	**Scratchpad 22 (scratchpad22):** Scratchpad bit
21	0h RW	**Scratchpad 21 (scratchpad21):** Scratchpad bit
20	0h RW	**Scratchpad 20 (scratchpad20):** Scratchpad bit
19	0h RW	**Scratchpad 19 (scratchpad19):** Scratchpad bit
18	0h RW	**Scratchpad 18 (scratchpad18):** Scratchpad bit
17	0h RW	**Scratchpad 17 (scratchpad17):** Scratchpad bit
16	0h RW	**Scratchpad 16 (scratchpad16):** Scratchpad bit
15	0h RW	**Scratchpad 15 (scratchpad15):** Scratchpad bit
14	0h RW	**Scratchpad 14 (scratchpad14):** Scratchpad bit
13	0h RW	**Scratchpad 13 (scratchpad13):** Scratchpad bit
12	0h RW	**Scratchpad 12 (scratchpad12):** Scratchpad bit
11	0h RW	**Scratchpad 11 (scratchpad11):** Scratchpad bit
10	0h RW	**Scratchpad 10 (scratchpad10):** Scratchpad bit
9	0h RW	**Scratchpad 9 (scratchpad9):** Scratchpad bit
8	0h RW	**Scratchpad 8 (scratchpad8):** Scratchpad bit
7	0h RW	**Scratchpad 7 (scratchpad7):** Scratchpad bit
6	0h RW	**Scratchpad 6 (scratchpad6):** Scratchpad bit
5	0h RW	**Scratchpad 5 (scratchpad5):** Scratchpad bit
4	0h RW	**U2U Drive Host Bypass Clock Ready Gate Enable (scratchpad4):** Used to prevent the clk_in_ready from U_DWC_U2UB_clk_mux_host from enabling U_DWC_U2UB_clkgate_cell_hostclk_in clock gating cell

Bit Range	Default & Access	Field Name (ID): Description
3	1h RW	**U2U Drive Dev Bypass Clock Ready Gate Enable (scratchpad3):** Used to prevent the clk_in_ready from U_DWC_U2UB_clk_mux_dev from enabling U_DWC_U2UB_clkgate_cell_devclk_in clock gating cell
2	0h RW	**Disable CNVI SX Wake (scratchpad2):** When set, PMC FW will send ForcePwrGatePOK and assert cnv_pgcb_rst on SX Entry
1	0h RW	**S0IX Inhibit (s0ix_inhibit):** Scratchpad bit used by PCIE ASL code
0	0h RW	**MOT Enable (mot_enable):** Indicates that MOT is enabled. When set to '1', PMC firmware will only switch ART to RTC after receiving the VnnOffPrepAck from pcode. This ensures proper time sync handling all the way until PCS is down.

10.107 Display Hot Plug Detect Control (DISPLAY_HPD_CTL)—Offset 1094h

Controls for PMC display hot plug detect logic. Hot plug detect is offloaded from display during S0ix, is used to gate S0ix entry, is an S0ix wake condition, and may result in notification to display controller. PMC implements a simplified hot plug detect, where the input is deglitched and edges are detected.

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: F0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (rsvd):** Reserved.
15:13	0h RW	**Edge Rising (edge_rising):** Enable detection of rising edges. One bit per GPIO HPD wire. More than one GPIO can be enabled. One or both edges may be of interest, and the edge of interest may change based on polarity of GPIO HPD wire. edge_rising = 3'b001: ddi0 rising edges enabled edge_rising = 3'b010: ddi1 rising edges enabled ... edge_rising = 3'b111: ddi2, ddi1 and ddi0 rising edges enabled
12:10	0h RW	**Edge Falling (edge_falling):** Enable detection of falling edges. One bit per GPIO HPD wire. More than one GPIO can be enabled. One or both edges may be of interest, and the edge of interest may change based on polarity of GPIO HPD wire. edge_falling = 3'b001: ddi0 falling edges enabled edge_falling = 3'b010: ddi1 falling edges enabled ... edge_falling = 3'b111: ddi2, ddi1 and ddi0 falling edges enabled

Bit Range	Default & Access	Field Name (ID): Description
9:5	7h RW	**Deglitch (deglitch):** Deglitch time, in 60 us steps. HPD logic will ignore any pulses shorter than the deglitch time. A transition which is held longer than the deglitch time will be considered an edge, and will result in an HPD event, which may prevent S0ix entry or wake from S0ix and notify display controller. All settings have an error of +/- 1 tick (60 us). 4'd0: 000 us (invalid configuration) 5'd1: 60 us (invalid configuration) 5'd2: 120 us ... 5'd7: 420 us (default) ... 5'd31: 1.86 ms
4:0	10h RW	**Recent Edge Detect (edge_det_rcnt):** Recent event time, in 10 ms steps. HPD logic will hold a wire asserted for a programmable period of time after an enabled edge has been detected, in order to prevent S0ix entry for this period of time. All settings have an error of +/- 1 tick (10 ms). Default: 160 ms Max: 310 ms

10.108 U2U Bridge Survivability Register 1 (U2UB_SR_1)—Offset 10A0h

This register contains survivability bits for the U2U bridge IP. The bits are read and writeable by BIOS, and their values are sent by direct wire to the U2U.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**rsvd:** RESERVED
29:28	0h RW	**force_host_opmode_value:** Override value for UTMI host opmode when corresponding override enable bit is set
27	0h RW	**force_host_opmode:** Override enable for UTMI host opmode
26:25	0h RW	**drive_host_linestate_value:** Override value for UTMI host line-state when corresponding override enable bit is set
24	0h RW	**drive_host_linestate:** Override enable for UTMI host line-state
23	0h RW	**clr_host_fifo:** Override bit to clear the host Tx fifo
22	0h RW	**drive_utmiclk_req_host_value:** Override value for host utmiclk request when corresponding override enable bit is set

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**drive_utmiclk_req_host:** Override enable for host utmiclk request
20	0h RW	**assert_dev_u2u_utmi_reset:** Override the device to U2U UTMI reset
19	0h RW	**drive_dev_clk_mux_sel_value:** Override value for device clock mux select when corresponding override enable bit is set. 1 indicates aux clock, and 0 indicates utmi clock.
18	0h RW	**drive_dev_clk_mux_sel:** Override enable for device clock mux select to choose between aux clock and device utmi clock
17	0h RW	**drive_gate_dev_clk_value:** Override value for device clock gating when corresponding override enable bit is set. 1 indicates CG enabled, and 0 indicates CG disabled.
16	0h RW	**drive_gate_dev_clk:** Override enable for device clock gating
15:14	0h RW	**force_dev_xcvrsel_value:** Override value for UTMI device xcvrselect when corresponding override enable bit is set
13	0h RW	**force_dev_xcvrsel:** Override enable for UTMI device xcvrselect
12:11	0h RW	**force_dev_opmode_value:** Override value for UTMI device opmode when corresponding override enable bit is set
10	0h RW	**force_dev_opmode:** Override enable for UTMI device opmode
9:8	0h RW	**drive_dev_linestate_value:** Override value for UTMI device line-state when corresponding override enable bit is set
7	0h RW	**drive_dev_linestate:** Override enable for UTMI device line-state
6	0h RW	**clr_dev_fifo:** Override bit to clear the device Tx fifo
5	0h RW	**drive_utmiclk_req_dev_value:** Override value for device utmiclk request when corresponding override enable bit is set
4	0h RW	**drive_utmiclk_req_dev:** Override enable for device utmiclk request
3	0h RW	**drive_dev_vbusvalid_value:** Override value for device vbusvalid when corresponding override enable bit is set
2	0h RW	**drive_dev_vbusvalid:** Override enable for device vbusvalid
1	0h RW	**drive_auxclk_req_value:** Override value for auxclk request when corresponding override enable bit is set
0	0h RW	**drive_auxclk_req:** Override enable for auxclk request

10.109 U2U Bridge Survivability Register 2 (U2UB_SR_2)—Offset 10A4h

This register contains survivability bits for the U2U bridge IP. The bits are read and writeable by BIOS, and their values are sent by direct wire to the U2U.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 10h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**rsvd:** RESERVED
15	0h RW	**drive_hrx_pe_rx_sync_eop_val:** Override value for host rx packet emulator FSM control signal (rx_sync_eop) when corresponding override enable bit is set
14	0h RW	**drive_hrx_pe_rx_sync_eop:** Override enable for host rx packet emulator FSM control signal (rx_sync_eop)
13	0h RW	**drive_dtx_pe_utmi_txvalid_val:** Override value for device tx packet emulator FSM control signal (utmi_txvalid) when corresponding override enable bit is set
12	0h RW	**drive_dtx_pe_utmi_txvalid:** Override enable for device tx packet emulator FSM control signal (utmi_txvalid)
11	0h RW	**drive_drx_pe_rx_sync_eop_val:** Override value for device rx packet emulator FSM control signal (rx_sync_eop) when corresponding override enable bit is set
10	0h RW	**drive_drx_pe_rx_sync_eop:** Override enable for device rx packet emulator FSM control signal (rx_sync_eop)
9	0h RW	**drive_htx_pe_utmi_txvalid_val:** Override value for host tx packet emulator FSM control signal (utmi_txvalid) when corresponding override enable bit is set
8	0h RW	**drive_htx_pe_utmi_txvalid:** Override enable for host tx packet emulator FSM control signal (utmi_txvalid)
7	0h RW	**assert_host_u2u_utmi_reset:** Override the host to U2U UTMI reset
6	0h RW	**drive_host_clk_mux_sel_value:** Override value for host clock mux select when corresponding override enable bit is set. 1 indicates aux clock, and 0 indicates utmi clock.
5	0h RW	**drive_host_clk_mux_sel:** Override enable for host clock mux select to choose between aux clock and host utmi clock
4	1h RW	**drive_gate_host_clk_value:** Override value for host clock gating when corresponding override enable bit is set. 1 indicates CG enabled, and 0 indicates CG disabled.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**drive_gate_host_clk:** Override enable for host clock gating
2:1	0h RW	**force_host_xcvrsel_value:** Override value for UTMI host xcvrselect when corresponding enable bit is set
0	0h RW	**force_host_xcvrsel:** Override enable for UTMI host xcvrselect

10.110 OBFF Control and Status (OBFF_CTL_STS)—Offset 10C8h

Through this register, BIOS controls the External OBFF feature through this register and, PMC indicates that it is currently processing an OBFF state change. The timing parameters for GPIO WAKE pin indication are also present in this register.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 6450280Fh

Bit Range	Default & Access	Field Name (ID): Description
31:24	64h RW/L	**Tobff3 (Tobff3):** Minimum time between two consecutive state changes (to avoid misinterpretation of pattern). Programmable timer to control Tobff3. Timer counts 10 ns clock cycles. Tobff3(min) = 700 ns Default = 0x64 = 100 clocks = 1000 ns
23:16	50h RW/L	**Tobff2 (Tobff2):** Time between two falling WAKE# edges when signaling CPU Active. Programmable timer to control Tobff2. Timer counts 10 ns clock cycles. Tobff2(min) = 700 ns Tobff2(max) = 1000 ns Default = 0x50 = 80 clocks = 800 ns Note: Tobff2 must be programmed to a value at least 2x of Tobff1.
15:8	28h RW/L	**Tobff1 (Tobff1):** Minimum/Maximum WAKE# pulse width; applies to both active-inactive-active and inactive-active-inactive cases. Programmable timer to control Tobff1. Timer counts 10 ns clock cycles. Tobff1(min) = 300 ns Tobff1(max) = 500 ns Default = 0x28 = 40 clocks = 400 ns
7	0h RO/V	**PORT3 OBFF Status (PORT3_OBFF_STATUS):** OBFF Status: Indicates if PMC is currently processing an OBFF state change. ASL code will read this bit on RTD3 Entry to ensure no conflicts on WAKE# GPIO as part of RTD3 flows.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RO/V	**PORT2 OBFF Status (PORT2_OBFF_STATUS):** OBFF Status: Indicates if PMC is currently processing an OBFF state change. ASL code will read this bit on RTD3 Entry to ensure no conflicts on WAKE# GPIO as part of RTD3 flows.
5	0h RO/V	**PORT1 OBFF Status (PORT1_OBFF_STATUS):** OBFF Status: Indicates if PMC is currently processing an OBFF state change. ASL code will read this bit on RTD3 Entry to ensure no conflicts on WAKE# GPIO as part of RTD3 flows.
4	0h RO/V	**PORT0 OBFF Status (PORT0_OBFF_STATUS):** OBFF Status: Indicates if PMC is currently processing an OBFF state change. ASL code will read this bit on RTD3 Entry to ensure no conflicts on WAKE# GPIO as part of RTD3 flows.
3	1h RW/L	**PORT3 OBFF Disable (PORT3_OBFF_DISABLE):** OBFF Disable: When set, disables PMC from driving external OBFF signaling to the PCIe devices through the WAKE GPIO#.OBFF must only be driven to PCIe device when in D0. ASL code will provide this control by setting and clearing this bit on RTD3 Entry/Exit FlowsIf set by BIOS the named function is exposed to the OS via ACPI
2	1h RW/L	**PORT2 OBFF Disable (PORT2_OBFF_DISABLE):** OBFF Disable: When set, disables PMC from driving external OBFF signaling to the PCIe devices through the WAKE GPIO#.OBFF must only be driven to PCIe device when in D0. ASL code will provide this control by setting and clearing this bit on RTD3 Entry/Exit FlowsIf set by BIOS the named function is exposed to the OS via ACPI
1	1h RW/L	**PORT1 OBFF Disable (PORT1_OBFF_DISABLE):** OBFF Disable: When set, disables PMC from driving external OBFF signaling to the PCIe devices through the WAKE GPIO#.OBFF must only be driven to PCIe device when in D0. ASL code will provide this control by setting and clearing this bit on RTD3 Entry/Exit FlowsIf set by BIOS the named function is exposed to the OS via ACPI
0	1h RW/L	**PORT0 OBFF Disable (PORT0_OBFF_DISABLE):** OBFF Disable: When set, disables PMC from driving external OBFF signaling to the PCIe devices through the WAKE GPIO#.OBFF must only be driven to PCIe device when in D0. ASL code will provide this control by setting and clearing this bit on RTD3 Entry/Exit FlowsIf set by BIOS the named function is exposed to the OS via ACPI

10.111 Lock Register (LOCK)—Offset 10CCh

Register | Field | Lock bit ACPI.SMI_EN | espi_smi_en | LOCK.espi_smi
OBFF_CTL_STS | See description | LOCK.obff
PMC_CFG2 | pwrbtn_dis | LOCK.pwrbtn
GEN_PMCON2 | lpc_lpb_clk_ctrl | LOCK.lpc_lpb_clk_ctrl
GEN_PMCON3 | See description | LOCK.slpsx_str_pol_lock
PM_CFG | See description | LOCK.slpsx_str_pol_lock
IRQ_SEL_0 | All | LOCK.irq_sel
IRQ_SEL_1 | All | LOCK.irq_sel
IRQ_SEL_2 | All | LOCK.irq_sel
CRID | All | LOCK.crid

GEN_PMCON1 | pme_b0_s5_dis | LOCK.sx_wake
GEN_PMCON1 | ag3e | LOCK.sx_wake
RSVD[GEN_PMCON2] | bios_pci_exp_en | LOCK.pcie
GEN_PMCON2 | per_smi_sel | LOCK.per_smi
FUNC_DIS_0 | All | LOCK.func_dis
FUNC_DIS_1 | All | LOCK.func_dis
S0IX_WAKE_EN | All | LOCK.s0ix
ETR | MAX_S0IX | LOCK.s0ix
IRQ_CFG_UART | All | LOCK.irq_cfg_uart

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**Reserved (rsvd):** Reserved.
13	0h RW/L	**ESPI SMI Lock (espi_smi):** When this bit is set, writes to the acpi.smi_en.ESPI_SMI_EN bit will have no effect. Once the LOCK.ESPI_SMI bit is set, writes of 0 to the LOCK.ESPI_SMI bit will have no effect.
12	0h RW/L	**OBFF Lock (obff):** Locks GCR.OBFF_CTL_STS.Tobff1, Tobff2, Tobff3, PORT0_OBFF_DISABLE, PORT1_OBFF_DISABLE, PORT2_OBFF_DISABLE, PORT3_OBFF_DISABLE. This field is also self-locking.
11	0h RW/L	**Power Button Lock (pwrbtn):** Locks GCR.PMC_CFG2.pwrbtn_dis. This field is also self-locking.
10	0h RW/L	**LPC Loopback Clock Control Lock (lpc_lpb_clk_ctrl):** Locks GCR.GEN_PMCON2.lpc_lpb_clk_ctrl. This field is also self-locking.
9	0h RW/L	**ASLP Sx# Stretching Policy Lock-Down (slpsx_str_pol_lock):** When set to 1, this bit locks down the following fields: GEN_PMCON3.DIS_SLP_X_STRCH_SUSPF GEN_PMCON3.SLP_S3_MIN_ASST_WDTH GEN_PMCON3.S4MAW GEN_PMCON3.S4ASE PM_CFG.SLP_A_MIN_ASST_WDTH (if applicable) PM_CFG.SLP_LAN_MIN_ASST_WDTH (if applicable) PM_CFG.PWR_CYC_DUR Those bits become read-only. This bit becomes locked when a value of 1b is written to it. Writes of 0 to this bit are always ignored. Once locked by writing 1, the only way to clear this bit is to perform a platform reset. This lockdown bit is available in both desktop and mobile. Note: Relevant only for no PMIC mode.
8	0h RW/L	**IRQ Select Lock (irq_sel)**

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW/L	**CRID Lock (crid)**
6	0h RW/L	**SX Wake Lock (sx_wake)**
5	0h RO	**Reserved (rsvd_pcie):** Reserved: PCIE lock.
4	0h RW/L	**Periodic SMI Lock (per_smi)**
3	0h RW/L	**Func Dis Lock (func_dis)**
2	0h RW/L	**Reserved (s0ix):** Reserved.
1	0h RW/L	**UART IRQ Configuration Lock (irq_cfg_uart):** Locks the IRQ_CFG_UART register
0	0h RO	**Reserved (reserved0):** Reserved.

IPC Command (IPC_CMD)—Offset 0h

The IPC CMD register is used for conveying the type of commands to the IPC block. This is a special kind of register and a write to this register always results in Interrupt to the ARC and Setting of the busy bit in the IPC_STS register. The setting of busy bit in the IPC_STS register results in blocking of all transaction from the IA host. The ARC FW is expected to read the IPC_CMD register, whenever interrupted, interpret it and follow the instructions as ordered by the IA host. The IA host may optionally prefer to be interrupted, when the command is complete by sending an MSI. When the ARC FW is done with the command, it is expected to update the IPC_STS register and clear the busy bit.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**Reserved (Reserved0):** Reserved.
23:16	0h RW	**Size (size):** Size of the transaction in bytes. This 8-bit field can potentially specify 256 Bytes, but the hardware based write_buffer and read_buffer are both only 16 bytes deep. So the size should not be greater than 16bytes for normal reads and normal write operation. However, larger sizes can be used for indirect operations.
15:12	0h RW	**Command ID (cmd_ID):** ID or Tag associated with this command. When the IPC command is completed, this value is copied in the IPC_STS register by the hardware.

Bit Range	Default & Access	Field Name (ID): Description
11:9	0h RW	**Reserved (rsvd):** Reserved.
8	0h RW	**MSI (msi):** Send an MSI once the command is executed.
7:0	0h RW	**Command (Command):** Following are defined commands: • 00h: Normal Write • 01h: Message Write • 02h: Indirect Read • 03h: Reserved • 04h: Read DMA • 05h: Indirect write • 06h: FFh - Reserved

10.112 IPC Status (IPC_STS)—Offset 4h

The IPC status register provides information related to the status of the IPC block and also some information related to the last executed command.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (rsvd):** Reserved.
23:16	0h RW/V	**Error Code (error_code):** If the err bit is set, this field may contain some error code. ARC FW is responsible for updating this value.
15:8	0h RO/V	**Initiator ID (Initiator_ID):** ID of the initiator, whose command was executed last. The initiator_ID information is available through the CONNID OCP. This bits are captured by the IPC hardware and placed in this field. This information could be used by the ARC FW to figure out the initiator for any particular IPC command and whether to service it or not. It may also be used for debug purpose, when the ARC FW is unable to service some IPC command and sets the error bit in the IPC_STC register.
7:4	0h RO/V	**Command ID (cmd_ID):** ID of the last command. The PMU Hardware copies this same field from the IPC_CMD register and places it in this reg.
3	0h RO/V	**Reserved (rsvd1):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW/1C/V	**IRQ (irq):** This bit is set when FW clears IPC_STS.Busy if IPC_CMD.msi is set.
1	0h RW/V	**Error (Error):** There was some error in executing the last command. This value is written by the ARC FW.
0	0h RW/V	**Busy (Busy):** IPC block is busy executing some command. When done, this bit would be unset. This bit is set, whenever the IPC hardware receives a new command on IPC_CMD register. ARC FW is responsible for unsetting this bit. For IPC2, the IPC hardware blocks any new transactions from cDMI, to the IPC block until this bit is cleared.

10.113 IPC Source Pointer (IPC_SPTR)—Offset 8h

This is a 32-bit register containing the address of the read transaction. The IPC Source Pointer Register is used by the IA host to perform indirect reads. The IA host would first write to this IPC_SPTR register and specify the read address location. Next, the IA host would issue an Indirect Read command on the IPC_CMD register. The ARC FW is responsible to decode this transaction and issue a read on the address specified by the IPC_SPTR register.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Source Pointer (SPTR):** Read transaction source address.

10.114 IPC Destination Pointer (IPC_DPTR)—Offset Ch

This register contains the destination address of indirect read or indirect write transaction. Please refer to IPC_CMD register for detailed usage of this register.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Destination Pointer (DPTR):** Read or write transaction destination address.

10.115 IPC Write Buffer (IPC_WBUF0)—Offset 80h

The write buffer is 16 byte deep and is used by the IA host to write any data for ARC. This buffer is also used for writing Lincroft Msg and MsgD to system controller.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Write Buffer (WBUF)**

10.116 IPC Write Buffer (IPC_WBUF1)—Offset 84h

The write buffer is 16 byte deep and is used by the IA host to write any data for ARC. This buffer is also used for writing Lincroft Msg and MsgD to system controller.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Write Buffer (WBUF)**

10.117 IPC Write Buffer (IPC_WBUF2)—Offset 88h

The write buffer is 16 byte deep and is used by the IA host to write any data for ARC. This buffer is also used for writing Lincroft Msg and MsgD to system controller.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Write Buffer (WBUF)**

10.118 IPC Write Buffer (IPC_WBUF3)—Offset 8Ch

The write buffer is 16 byte deep and is used by the IA host to write any data for ARC. This buffer is also used for writing Lincroft Msg and MsgD to system controller.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Write Buffer (WBUF)**

10.119 IPC Read Buffer (IPC_RBUF0)—Offset 90h

This 16-byte deep read buffer is used by the IA host for gathering some data from the system controller. In the normal read mode, the IA host can ask the ARC to gather some information and store it in this IPC_RBUF buffer. Later on, the IA host can access this information.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Read Buffer (RBUF)**

10.120 IPC Read Buffer (IPC_RBUF1)—Offset 94h

This 16-byte deep read buffer is used by the IA host for gathering some data from the system controller. In the normal read mode, the IA host can ask the ARC to gather some information and store it in this IPC_RBUF buffer. Later on, the IA host can access this information.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Read Buffer (RBUF)**

10.121 IPC Read Buffer (IPC_RBUF2)—Offset 98h

This 16-byte deep read buffer is used by the IA host for gathering some data from the system controller. In the normal read mode, the IA host can ask the ARC to gather some information and store it in this IPC_RBUF buffer. Later on, the IA host can access this information.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Read Buffer (RBUF)**

10.122 IPC Read Buffer (IPC_RBUF3)—Offset 9Ch

This 16-byte deep read buffer is used by the IA host for gathering some data from the system controller. In the normal read mode, the IA host can ask the ARC to gather some information and store it in this IPC_RBUF buffer. Later on, the IA host can access this information.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Read Buffer (RBUF)**

PWM Control Register (PWMCTRL_0)—Offset 0h

PWMCTRL controls the PWM and needs to be accessed in particular order to prevent any blips. Below is the recommended flow for programming the PWMCTRL registers. Initial Enable or First Activation:

1. Program the Base Unit and On Time Divisor values.
2. Set the Software Update Bit.
3. Enable the PWM Output by setting PWM Enable.

4. Repeat the above steps for the next PWM Module.

Dynamic update while PWM is Enabled:

1. Program the Base Unit and On Time Divisor values.
2. Set the Software Update Bit.
3. Repeat the above steps for the next PWM module.

PWM output frequency = 19.2MHz * (Base_Unit / 256)

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 400000h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Enable (ENABLE):** Enable PWM output.
30	0h RW/V	**Software Update (SW_UPDATE):** Indication that there is an update to PWM settings pending. SW sets this bit to 1 when updating the Base Unit (Integer or Fraction) On-time Divisor. The PWM module will apply the new settings at the end of the current cycle and clear this bit.
29:22	1h RW	**Base Unit Integer (BASE_UNIT_INT):** Integer portion of Base Unit (Unsigned 8 integer bits and 14 fraction bits). Base_Unit = base_unit_int + (base_unit_frac / 16384)
21:8	0h RW	**Base Unit Frac (BASE_UNIT_FRAC):** Fractional portion of Base Unit (Unsigned 8 integer bits and 14 fraction bits). Base Unit = base_unit_int + (base_unit_frac / 16384)
7:0	0h RW	**On Time Divisor (ON_TIME_DIVISOR):** The On-time Divisor determines the PWM duty cycle. It is the number used to compare with the output of the base_unit counter.

10.123 PWM D0i3 Control Register (PWMD0i3C)—Offset 100h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 8h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (Rsvd):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
3	1h RW/1C/V	**Restore Required (rr)**
2	0h RW	**D0i3 (i3)**
1	0h RO	**Interrupt Request (ir)**
0	0h RO	**Command in Progress (cip)**

10.124 PWM Control Register (PWMCTRL_1)—Offset 400h

PWMCTRL controls the PWM and needs to be accessed in particular order to prevent any blips. Below is the recommended flow for programming the PWMCTRL registers. Initial Enable or First Activation:

1. Program the Base Unit and On Time Divisor values.
2. Set the Software Update Bit.
3. Enable the PWM Output by setting PWM Enable.
4. Repeat the above steps for the next PWM Module.

Dynamic update while PWM is Enabled:

1. Program the Base Unit and On Time Divisor values.
2. Set the Software Update Bit.
3. Repeat the above steps for the next PWM module.

PWM output frequency = 19.2MHz * (Base_Unit / 256)

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 400000h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Enable (ENABLE):** Enable PWM output.
30	0h RW/V	**Software Update (SW_UPDATE):** Indication that there is an update to PWM settings pending. SW sets this bit to 1 when updating the Base Unit (Integer or Fraction) On-time Divisor. The PWM module will apply the new settings at the end of the current cycle and clear this bit.
29:22	1h RW	**Base Unit Integer (BASE_UNIT_INT):** Integer portion of Base Unit (Unsigned 8 integer bits and 14 fraction bits). Base_Unit = base_unit_int + (base_unit_frac / 16384)

Bit Range	Default & Access	Field Name (ID): Description
21:8	0h RW	**Base Unit Frac (BASE_UNIT_FRAC):** Fractional portion of Base Unit (Unsigned 8 integer bits and 14 fraction bits). Base Unit = base_unit_int + (base_unit_frac / 16384)
7:0	0h RW	**On Time Divisor (ON_TIME_DIVISOR):** The On-time Divisor determines the PWM duty cycle. It is the number used to compare with the output of the base_unit counter.

10.125 PWM Control Register (PWMCTRL_2)—Offset 800h

PWMCTRL controls the PWM and needs to be accessed in particular order to prevent any blips. Below is the recommended flow for programming the PWMCTRL registers. Initial Enable or First Activation:

1. Program the Base Unit and On Time Divisor values.
2. Set the Software Update Bit.
3. Enable the PWM Output by setting PWM Enable.
4. Repeat the above steps for the next PWM Module.

Dynamic update while PWM is Enabled:

1. Program the Base Unit and On Time Divisor values.
2. Set the Software Update Bit.
3. Repeat the above steps for the next PWM module.

PWM output frequency = 19.2MHz * (Base_Unit / 256)

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 400000h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Enable (ENABLE):** Enable PWM output.
30	0h RW/V	**Software Update (SW_UPDATE):** Indication that there is an update to PWM settings pending. SW sets this bit to 1 when updating the Base Unit (Integer or Fraction) On-time Divisor. The PWM module will apply the new settings at the end of the current cycle and clear this bit.
29:22	1h RW	**Base Unit Integer (BASE_UNIT_INT):** Integer portion of Base Unit (Unsigned 8 integer bits and 14 fraction bits). Base_Unit = base_unit_int + (base_unit_frac / 16384)

Bit Range	Default & Access	Field Name (ID): Description
21:8	0h RW	**Base Unit Frac (BASE_UNIT_FRAC):** Fractional portion of Base Unit (Unsigned 8 integer bits and 14 fraction bits). Base Unit = base_unit_int + (base_unit_frac / 16384)
7:0	0h RW	**On Time Divisor (ON_TIME_DIVISOR):** The On-time Divisor determines the PWM duty cycle. It is the number used to compare with the output of the base_unit counter.

10.126 PWM Control Register (PWMCTRL_3)—Offset C00h

PWMCTRL controls the PWM and needs to be accessed in particular order to prevent any blips. Below is the recommended flow for programming the PWMCTRL registers. Initial Enable or First Activation:

1. Program the Base Unit and On Time Divisor values.
2. Set the Software Update Bit.
3. Enable the PWM Output by setting PWM Enable.
4. Repeat the above steps for the next PWM Module.

Dynamic update while PWM is Enabled:

1. Program the Base Unit and On Time Divisor values.
2. Set the Software Update Bit.
3. Repeat the above steps for the next PWM module.

PWM output frequency = 19.2MHz * (Base_Unit / 256)

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 400000h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Enable (ENABLE):** Enable PWM output.
30	0h RW/V	**Software Update (SW_UPDATE):** Indication that there is an update to PWM settings pending. SW sets this bit to 1 when updating the Base Unit (Integer or Fraction) On-time Divisor. The PWM module will apply the new settings at the end of the current cycle and clear this bit.
29:22	1h RW	**Base Unit Integer (BASE_UNIT_INT):** Integer portion of Base Unit (Unsigned 8 integer bits and 14 fraction bits). Base_Unit = base_unit_int + (base_unit_frac / 16384)

Bit Range	Default & Access	Field Name (ID): Description
21:8	0h RW	**Base Unit Frac (BASE_UNIT_FRAC):** Fractional portion of Base Unit (Unsigned 8 integer bits and 14 fraction bits). Base Unit = base_unit_int + (base_unit_frac / 16384)
7:0	0h RW	**On Time Divisor (ON_TIME_DIVISOR):** The On-time Divisor determines the PWM duty cycle. It is the number used to compare with the output of the base_unit counter.

§ §

11 Primary to Sideband Bridge Registers

This chapter documents the registers in Bus: 0, Device 13, Function 0.

11.1 PCI Identifier (PCIID) — Offset 0h

PCI header registers

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + 0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO/V	**Device Identification (DID):** The upper 8-bits of this field can be overridden by the SetID IOSF-SB Message.
15:0	0h RO	**Vendor Identification (VID):** Indicates Intel

11.2 PCI Command (PCICMD) — Offset 4h

PCI header registers

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:13, F:0] + 4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:11	0h RO	**Reserved (RSVD):** Reserved
10	0h RO	**Interrupt Disable (INTD):** P2SB does not issue any interrupts on its own behalf
9	0h RO	**Fast Back to Back Enable (FB2BE):** Not applicable
8	0h RW	**SERR# Enable (SEE):** this will enable parity error reporting to IEH
7	0h RO	**Reserved (RSVD_1):** Reserved
6	0h RW	**Parity Error Response Enable (PEE):** This bit controls the device's response to parity error.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RO	**VGA:** Not applicable.
4	0h RO	**Memory Write & Invalidate Enable (MWIE):** Not applicable.
3	0h RO	**Special Cycle Enable (SCE):** Not applicable.
2	0h RO	**Bus Master Enable (BME):** Bus mastering cannot be disabled as this device acts as a proxy for non-PCI devices.
1	0h RW	**Memory Space Enable (MSE):** Will control the P2SB acceptance of PCI MMIO BARs only. Other legacy regions are unaffected by this bit.
0	0h RW	**I/O Space Enable (IOSE):** Legacy regions are unaffected by this bit.

11.3 PCI Status (PCISTS) — Offset 6h

The P2SB does not issue any interrupts, signal any errors or provide any additional capability structures, so this register has no functionality.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:13, F:0] + 6h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**Detected Parity Error (DPE):** This register bit is set when parity error is detected.
14	0h RO	**Signaled System Error (SSE):** This register will set when PCICMD.SE bit and PCISTS.DPE bit is set
13:0	0h RO	**Reserved (RSVD):** Reserved

11.4 Revision ID (PCIRID) — Offset 8h

PCI header registers

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:13, F:0] + 8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO/V	**Revision ID (RID):** Indicates the part revision. This will reset to 0 but will overridden by the SetID IOSFSB message during the power-up sequence

11.5 Class Code (PCICC) — Offset 9h

PCI header registers

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + 9h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23:16	0h RO	**Base Class Code (BCC):** Indicates a memory controller device class.
15:8	0h RO	**Sub-Class Code (SCC):** Indicates an unspecified other memory controller.
7:0	0h RO	**Programming Interface (PI):** No programming interface.

11.6 PCI Master Latency Timer (PCIMLT) — Offset Dh

PCI header registers

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:13, F:0] + Dh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Reserved (RSVD):** Reserved

11.7 PCI Header Type (PCIHTYPE) — Offset Eh

PCI header registers

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:13, F:0] + Eh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**Multi-Function Device (MFD):** Indicates that this is part of a multi-function device.
6:0	0h RO	**Header Type (HTYPE):** Indicates a generic device header.

11.8 Sideband Register Access BAR (SBREG_BAR) — Offset 10h

PCI header registers

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + 10h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**Register Base Address (RBA):** Lower DWORD of the base address for the sideband register access BAR.
23:4	0h RO	**Hardwired to 0 to request a BAR of 16MB (HW202RB16MB):** Hardwired to 0 to request a BAR of 16MB
3	0h RO	**PREF:** Indicates this is not prefetchable.
2:1	0h RO	**Address Type (ATYPE):** Indicates that this can be placed anywhere in 64b space.
0	0h RO	**Space Type (STYPE):** Indicates memory space

11.9 Sideband Register BAR High DWORD (SBREG_BARH) — Offset 14h

PCI header registers

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + 14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Register Base Address (RBAH):** Upper DWORD of the base address for the sideband register access BAR.

11.10 PCI Subsystem Identifiers (PCIHSS) — Offset 2Ch

PCI header registers

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**Subsystem ID (SSID):** Written by BIOS. Not used by hardware.
15:0	0h RW/O	**Subsystem Vendor ID (SSVID):** Written by BIOS. Not used by hardware.

11.11 VLW Bus:Device:Function (VBDF) — Offset 50h

This register specifies the bus:device:function ID that the VLW VDM will use for its Requester ID. This field is default to Bus 0: Device 31: Function 0 after reset. BIOS shall program this field accordingly if unique bus:device:function number is required for the VLW message.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:13, F:0] + 50h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RW	**BUS:** VLW Bus Number
7:3	0h RW	**DEV:** VLW Device Number
2:0	0h RW	**FUNC:** VLW Function Number

11.12 ERROR Bus:Device:Function (EBDF) — Offset 52h

This register specifies the bus:device:function ID that the Error Signalling messages VDM will use for its Requester ID. This field is default to Bus 0: Device 31: Function 0 after reset. BIOS shall program this field accordingly if unique bus:device:function number is required for the VLW message.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:13, F:0] + 52h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RW	**BUS:** ERROR Bus Number
7:3	0h RW	**DEV:** ERROR Device Number
2:0	0h RW	**FUNC:** ERROR Function Number

11.13 Routing Configuration (RCFG) — Offset 54h

This register contains information used for routing transactions between primary and sideband interfaces.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + 54h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Reserved (RSVD):** Reserved
15:8	0h RW	**Reserved Page Register Destination ID (RPRID):** Specifies the IOSF-SB destination ID for sending Reserved Page Register cycles (e.g. Port 80h). By default this will load to the ID of the LPC or eSPI device depending on which has been strapped active in the system.
7:1	0h RW	**Reserved (RSVD_1):** Reserved
0	0h RW	**RTC Shadow Enable (RSE):** When set, all IO writes to the RTC will be also sent to the PMC. This allows cases where the battery backed storage is in an external PMIC.

11.14 High Performance Event Timer Configuration (HPTC) — Offset 60h

HPET configuration register

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:13, F:0] + 60h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**Address Enable (AE):** When set, the P2SB will decode the High Performance Timer memory address range selected by bits 1:0 below.
6:2	0h RO	**Reserved (RSVD):** Reserved
1:0	0h RW	**Address Select (AS):** This 2-bit field selects 1 of 4 possible memory address ranges for the High Performance Timer functionality. The encodings are: 00 : FED0_0000h - FED0_03FFFh 01 : FED0_1000h - FED0_13FFFh 10 : FED0_2000h - FED0_23FFFh 11 : FED0_3000h - FED0_33FFFh

11.15 IOxAPIC Configuration (IOAC) — Offset 64h

IOAPIC configuration register

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:13, F:0] + 64h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:9	0h RO	**Reserved (RSVD):** Reserved
8	0h RW	**Address Enable (AE):** When set, the P2SB will decode the IOxAPIC memory address range selected by bits 7:0 below.
7:0	0h RW	**APIC Range Select (ASEL):** These bits define address bits 19:12 for the IOxAPIC range. The default value of 00h enables compatibility with prior products as an initial value. This value must not be changed unless the IOxAPIC Enable bit is cleared.

11.16 IOxAPIC Bus:Device:Function (IBDF) — Offset 6Ch

This register specifies the bus:device:function ID that the IOxAPIC will use in the following : As the Requester ID when initiating Interrupt Messages to the CPU As the Completer ID when responding to the reads targeting the IOxAPICs Memory-Mapped I/O registers This field is default to Bus 0: Device 31: Function 0 after reset. BIOS shall program this field accordingly if unique bus:device:function number is required for the internal IOxAPIC.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:13, F:0] + 6Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RW	**BUS:** IOxAPIC Bus Number
7:3	0h RW	**DEV:** IOxAPIC Device Number
2:0	0h RW	**FUNC:** IOxAPIC Function Number

11.17 HPET Bus:Device:Function (HBDF) — Offset 70h

This register specifies the bus:device:function ID that the HPET device will use in the following : As the Requester ID when initiating Interrupt Messages to the CPU As the Completer ID when responding to the reads targeting the corresponding HPETs Memory-Mapped I/O registers This field is default to Bus 0: Device 31: Function 0 after reset. BIOS shall program this field accordingly if unique bus:device:function number is required for the corresponding HPET.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:13, F:0] + 70h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	**BUS:** HPET Bus Number
7:3	0h RO	**DEV:** HPET Device Number
2:0	0h RO	**FUNC:** HPET Function Number

11.18 Sideband Register posted 0 (SBREGPOSTED0) — Offset 80h

provides an mechanism to send MMIO writes as posted writes on the IOSFSB space

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + 80h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Sideband Register posted 0 (SBREGPOSTED0):** One hot masks for setting SBREG to posted posted for IOSF-SB endpoint IDs 31-0.

11.19 Sideband Register posted 1 (SBREGPOSTED1) — Offset 84h

provides an mechanism to send MMIO writes as posted writes on the IOSFSB space

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + 84h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Sideband Register posted 1 (SBREGPOSTED1):** One hot masks for setting SBREG to posted posted for IOSF-SB endpoint IDs 63-32.

11.20 Sideband Register posted 2 (SBREGPOSTED2) — Offset 88h

provides an mechanism to send MMIO writes as posted writes on the IOSFSB space

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + 88h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Sideband Register posted 0 (SBREGPOSTED2):** One hot masks for setting SBREG to posted posted for IOSF-SB endpoint IDs 95-64.

11.21 Sideband Register posted 3 (SBREGPOSTED3) — Offset 8Ch

provides an mechanism to send MMIO writes as posted writes on the IOSFSB space

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + 8Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Sideband Register posted 3 (SBREGPOSTED3):** One hot masks for setting SBREG to posted posted for IOSF-SB endpoint IDs 127-96.

11.22 Sideband Register posted 4 (SBREGPOSTED4) — Offset 90h

provides an mechanism to send MMIO writes as posted writes on the IOSFSB space

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + 90h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Sideband Register posted 4 (SBREGPOSTED4):** One hot masks for setting SBREG to posted posted for IOSF-SB endpoint IDs 159-128.

11.23 Sideband Register posted 5 (SBREGPOSTED5) — Offset 94h

provides an mechanism to send MMIO writes as posted writes on the IOSFSB space

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + 94h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Sideband Register posted 5 (SBREGPOSTED5):** One hot masks for setting SBREG to posted posted for IOSF-SB endpoint IDs 191-160.

11.24 Sideband Register posted 6 (SBREGPOSTED6) — Offset 98h

provides an mechanism to send MMIO writes as posted writes on the IOSFSB space

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + 98h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Sideband Register posted 6 (SBREGPOSTED6):** One hot masks for setting SBREG to posted posted for IOSF-SB endpoint IDs 223-192.

11.25 Sideband Register posted 7 (SBREGPOSTED7) — Offset 9Ch

provides an mechanism to send MMIO writes as posted writes on the IOSFSB space

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + 9Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Sideband Register posted 7 (SBREGPOSTED7):** One hot masks for setting SBREG to posted posted for IOSF-SB endpoint IDs 255-224.

11.26 Display Bus:Device:Function (DISPBDF) — Offset A0h

This register specifies the bus:device:function ID that the Display initiated upstream RAVDMs will use for its Requester ID. This will also be used for the claiming these Route-by-ID RAVDMs downstream.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + A0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD):** Reserved
18:16	0h RW	**Display Target Block (DTBLK):** This register contains the Target BLK field that will be used when sending RAVDM messages to the CPU Complex North Display.
15:8	0h RW	**BUS:** The bus number of the Display in the CPU Complex.
7:3	0h RW	**DEV:** The bus number of the Display in the CPU Complex.
2:0	0h RW	**FUNC:** The function number of the Display in the CPU Complex

11.27 ICC Register Offsets (ICCOS) — Offset A4h

This register contains the offsets to be used when sending RAVDMs to the Integrated Clock Controller. Each of the two spaces decoded for the ICC have a separate base address that will be used when sending those transactions on IOSF-SB to the ICC.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:13, F:0] + A4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RW	**Modulator Control Address Offset (MODBASE):** This specifies the upper 8b for the 16b address that will be used for sending RAVDM access that target the Modulator Control range of the ICC (FFF00h - FFFFFh).
7:0	0h RW	**Buffer Address Offset (BUFBASE):** This specifies the upper 8b for the 16b address that will be used for sending RAVDM access that target the Buffer range of the ICC (FFE00h - FFEFFh).

11.28 Endpoint Mask 0 (EPMASK0) — Offset B0h

provide a mechanism for disabling particular IOSF-SB endpoints from being allowed to targeted by transactions from the P2SB

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + B0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V/L	**Endpoint Mask 0 (EPMASK0):** One hot masks for disabling IOSF-SB endpoint IDs 31-0.

11.29 Endpoint Mask 1 (EPMASK1) — Offset B4h

provide a mechanism for disabling particular IOSF-SB endpoints from being allowed to targeted by transactions from the P2SB

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + B4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V/L	**Endpoint Mask 1 (EPMASK1):** One hot masks for disabling IOSF-SB endpoint IDs 63-32.

11.30 Endpoint Mask 2 (EPMASK2) — Offset B8h

provide a mechanism for disabling particular IOSF-SB endpoints from being allowed to targeted by transactions from the P2SB

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + B8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V/L	**Endpoint Mask 2 (EPMASK2):** One hot masks for disabling IOSF-SB endpoint IDs 95-64

11.31 Endpoint Mask 3 (EPMASK3) — Offset BCh

provide a mechanism for disabling particular IOSF-SB endpoints from being allowed to targeted by transactions from the P2SB

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + BCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V/L	**Endpoint Mask 3 (EPMASK3):** One hot masks for disabling IOSF-SB endpoint IDs 127-96

11.32 Endpoint Mask 4 (EPMASK4) — Offset C0h

provide a mechanism for disabling particular IOSF-SB endpoints from being allowed to targeted by transactions from the P2SB

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + C0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V/L	**Endpoint Mask 4 (EPMASK4):** One hot masks for disabling IOSF-SB endpoint IDs 128-159

11.33 Endpoint Mask 5 (EPMASK5) — Offset C4h

provide a mechanism for disabling particular IOSF-SB endpoints from being allowed to targeted by transactions from the P2SB

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + C4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V/L	**Endpoint Mask 5 (EPMASK5):** One hot masks for disabling IOSF-SB endpoint IDs 191-160

11.34 Endpoint Mask 6 (EPMASK6) — Offset C8h

provide a mechanism for disabling particular IOSF-SB endpoints from being allowed to targeted by transactions from the P2SB

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + C8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V/L	**Endpoint Mask 6 (EPMASK6):** One hot masks for disabling IOSF-SB endpoint IDs 223-192

11.35 Endpoint Mask 7 (EPMASK7) — Offset CCh

provide a mechanism for disabling particular IOSF-SB endpoints from being allowed to targeted by transactions from the P2SB

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + CCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V/L	**Endpoint Mask 7 (EPMASK7):** One hot masks for disabling IOSF-SB endpoint IDs 255-224

11.36 SBI Address (SBIADDR) — Offset D0h

Provides mechanism to send message on IOSFSB;The SAI check is only required on RS field but due to tool limitation, the SAI check is applied in RDL on whole register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + D0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**Destination Port ID (DESTID):** The content of this register field is sent in the IOSF Sideband Message Register Access dest field.
23:20	0h RO	**Reserved (RSVD):** Reserved
19:16	0h RW	**Root Space (RS):** Destination IOSF-SB Root Space. *Note: This register may only be written during manufacturing test. P2SB will only accept writes to this register from transactions with a SAI equal to the SBI_RS_ACCESS_SAI parameter. This should be assigned to the SAI used by the functional test module (typically TAM) that will perform this register write on IOSF-P.
15:0	0h RW	**Address Offset (OFFSET):** Register address offset. The content of this register field is sent in the IOSF Sideband Message Register Access address(15:0) field.

11.37 SBI Data (SBIDATA) — Offset D4h

Provides mechanism to send message on IOSFSB

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + D4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**DATA:** The content of this register field is sent on the IOSF sideband Message Register Access data(31:0) field.

11.38 SBI Status (SBISTAT) — Offset D8h

Provides mechanism to send message on IOSFSB

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:13, F:0] + D8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RW	**OPCODE:** This is the Opcode sent in the IOSF sideband message.
7	0h RW	**POSTED:** When set to 1, the message will be send as a posted message instead of non-posted. This should only be used if the receiver is known to support posted operations for the specified operation.
6:3	0h RO	**Reserved (RSVD):** Reserved
2:1	0h RW/V	**RESPONSE:** 00 - Successful 01 - Unsuccessful / Not Supported 10 - Powered Down 11 - Multi-cast Mixed This register reflects the response status for the previously completed transaction. The value of this register is only meaningful if SBISTAT.INITRDY is zero.
0	0h RW/1S	**Initiate/ Ready# (INITRDY):** 0: The IOSF sideband interface is ready for a new transaction 1: The IOSF sideband interface is busy with the previous transaction. A write to set this register bit to 1 will trigger an IOSF sideband message on the private IOSF sideband interface. The message will be formed based on the values programmed in the Sideband Message Interface Register Access registers. Software needs to ensure that the interface is not busy (SBISTAT.INITRDY is clear) before writing to this register.

11.39 SBI Routing Identification (SBIRID) — Offset DAh

Provides mechanism to send message on IOSFSB

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:13, F:0] + DAh	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:12	0h RW	**First Byte Enable (FBE):** The content of this field is sent in the IOSF Sideband Register Access FBE field.
11	0h RO	**Reserved (RSVD):** Reserved
10:8	0h RW	**Base Address Register (BAR):** The contents of this field are sent in the IOSF Sideband Register Access BAR field. This should be zero performing a Memory Mapped operation to a PCI compliant device.
7:0	0h RW	**Function ID (FID):** The contents of this field are sent in the IOSF Sideband Register access FID field. This field should generally remain at zero unless specifically required by a particular application.

11.40 SBI Extended Address (SBIEXTADDR) — Offset DCh

Provides mechanism to send message on IOSFSB

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + DCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Extended Address (ADDR):** The content of this register field is sent on the IOSF sideband Message Register Access address(48:32) field. This must be set to all 0 if 16b addressing is desired.

11.41 P2SB Control (P2SBC) — Offset E0h

P2SB general configuration register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + E0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/O	**SBI register Lock (SBILOCK):** Once written, it will not be writeable until module reset. Write once (1 or 0) to lock the Lock Bit. SBISTAT.INITRDY Register bit only lock if SBILOCK Bit = 1
30:27	0h RW	**Reserved (RSVD):** Reserved
26	0h RW	**Data Parity Error Enable (DPEE):** IOSFP data parity error handling(EP bit) enabling bit
25	0h RW	**Command Parity Error Enable (CPEE):** IOSFP command parity error handling enabling bit
24	0h RW	**Data Phase Parity Error Enable (DPPEE):** IOSFP data phase parity error handling enabling bit
23:18	0h RW	**Reserved (RSVD_3):** Reserved
17	0h RW/O	**Endpoint Mask Lock (MASKLOCK):** Locks the value of the EPMASK[0-7] registers. Once this value is written to a one it may only be cleared by a reset.
16	0h RW	**PGCB Clock Gating Enable (PGCBCGE):** When asserted the P2SB can de-assert the clock request to disable the PGCB clock dynamically when it reaches the power down idle state.
15:9	0h RW	**Reserved: (RSVD_1):** Reserved
8	0h RW	**HIDE:** When this bit is set, the P2SB will return 1s on any PCI Configuration Read on IOSF-P. All other transactions including PCI Configuration Writes are unaffected by this. This does not affect reads performed on the IOSF-SB interface.
7:3	0h RW	**Reserved (RSVD_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
2:0	0h RW	**Max Writes Pending (MAXW):** This controls the max number of outstanding writes on IOSF-SB initiated by MMIO writes to the SBREG_BAR. Once the number of SBREG_BAR writes issued but not completed on IOSF-SB is equal to this value no new requests will be forwarded until completions are received. A value of zero will have the same behavior as the value of one (single write outstanding).

11.42 Power Control Enable (PCE) — Offset E4h

Power Control Enable register

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:13, F:0] + E4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:6	0h RO	**Reserved (RSVD):** Reserved
5	0h RW	**Hardware Autonomous Enable (HAE):** When set, the P2SB will automatically engage power gating when it has reached its idle condition.
4:3	0h RO	**Reserved (RSVD_1):** Reserved
2	0h RO	**D3-Hot Enable (D3HE):** No support for D3 Hot power gating.
1	0h RO	**I3 Enable (I3E):** No support for S0i3 power gating.
0	0h RW	**PMC Power Gating Enable (PMCPG_EN):** When set to 1, the P2SB will engage power gating if it is idle and the pmc_p2sb_sw_pg_req_b signal is asserted.

11.43 Power Gate Control Block Timing (PGCBT) — Offset E6h

PGCB configuration register

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:13, F:0] + E6h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:12	0h RW	**Reserved (RSVD):** Reserved
11:10	0h RW	**Primary Reset De-assert to Next State Time (TIRSTUP):** Value representing the number of delay clocks required between the deassertion of pgcb_force_prim_rst_b and the next state in the FSM (varies). 00: 1 clock 01: 2 clocks (default) 10: 8 clocks 11: 256 clocks
9:8	0h RW	**Isolation Deassertion Time (TDEISOLATE):** Value representing the minimum number of delay clocks required between the deassertion of pgcb_isol_en_b until the restore flow begins. 00: 1 clock 01: 2 clocks (default) 10: 8 clocks 11: 256 clocks

Bit Range	Default & Access	Field Name (ID): Description
7:6	0h RW	**Sleep De-assert to ISM Unlock Time (TSLPINACTV):** Value representing the minimum number of delay clocks required between the deassertion of pgcb_sleep to the deassertion of either pgcb_ip_*_lock_req_b from state retention or pgcb_isol_en_b from a non-retention state. 00: 1 clock 01: 2 clocks (default) 10: 8 clocks 11: 256 clocks
5:4	0h RW	**Reset to Power Down Request Time (TRESETACT):** Value representing the minimum number of delay clocks required between the assertion of pgcb_force_prim_rst_b (and pgcb_force_rst_b) to either the assertion of pgcb_pmc_pg_req_b for state retention or pgcb_sleep for non-retention. 00: 1 clock 01: 2 clocks (default) 10: 8 clocks 11: 256 clocks
3:2	0h RW	**Isolation Active to Reset Time (TISOLATE):** alue representing the minimum number of pgcb_clk cyles required between the assertion of pgcb_isol_en_b to the assertion of pgcb_force_prim_rst_b and pgcb_force_rst_b. 00: 1 clock 01: 2 clocks (default) 10: 8 clocks 11: 256 clocks
1:0	0h RW	**Sleep to Isolation Active Time (TSLEEPACT):** Value representing the minimum number of pgcb_clk cyles required between the assertion of pgcb_sleep (and the deassertion of pgcb_isol_latchen) to the assertion of pgcb_isol_en_b. 00: 1 clock 01: 2 clocks (default) 10: 8 clocks 11: 256 clocks

11.44 Primary Clock Domain Controls (PDOMAIN) — Offset E8h

Primary CDC configuration register

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:13, F:0] + E8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:12	0h RW	**Primary CLKREQ Hold-On (CKREQHO):** Defines the amount of idle time required between locking for power gate preparation and deassertion of the prim_clkreq signal. This field defines the exponent such that the actual delay = 2^ CKREQHO.
11:8	0h RW	**Primary Domain Lock Hold-Off (LOCKHO):** Defines the amount of idle time required between local clock gating and locking for power gate preparation (if power gating is enabled). This field defines the exponent such that the actual delay = 2^ LOCKHO.
7:4	0h RW	**Primary Clock Gating Hold-Off (CGATEHO):** Defines the amount of idle time required before local clock gating will be engaged (if enabled). This field defines the exponent such that the actual delay = 2^ CGATEHO.
3:2	0h RW	**Reserved (RSVD):** Reserved
1	0h RW	**Primary CLKREQ DISABLE (CKREQD):** Primary CLKREQ DISABLE
0	0h RW	**Primary Clock Gating Disable (CGD):** When set to 1, the local clock gating for the IOSF-P clock domain within the P2SB will be disabled.

11.45 Sideband Clock Domain Controls (SDOMAIN) — Offset EAh

Sideband CDC configuration register

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:13, F:0] + EAh	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:12	0h RW	**Sideband CLKREQ Hold-On (CKREQHO):** Defines the amount of idle time required between locking for power gate preparation and deassertion of the prim_clkreq signal. This field defines the exponent such that the actual delay = 2^ SCKREQHO.
11:8	0h RW	**Sideband Domain Lock Hold-Off (LOCKHO):** Defines the amount of idle time required between local clock gating and locking for power gate preparation (if power gating is enabled). This field defines the exponent such that the actual delay = 2^ PLOCKHO.
7:4	0h RW	**Sideband Clock Gating Hold-Off (CGATEHO):** Defines the amount of idle time required before local clock gating will be engaged (if enabled). This field defines the exponent such that the actual delay = 2^ CGATEHO.
3:2	0h RW	**Reserved (RSVD):** Reserved
1	0h RW	**Sideband CLKREQ DISABLE (CKREQD):** Sideband CLKREQ DISABLE
0	0h RW	**Sideband Clock Gating Disable (CGD):** When set to 1, the local clock gating for the IOSF-P clock domain within the P2SB will be disabled.

11.46 Unsupported Request Error Status (URES) — Offset F0h

This register is only reset by a loss of core power

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + F0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Reserved (RSVD):** Reserved
8	0h RW/1C	**Unsupported Request Detected (URD):** Set to 1 by hardware upon detecting an Unspported Request on IOSF Primary interface that is not considered Advisory Non-Fatal Used only when ENABLE_IEH is set
7:2	0h RO	**Reserved (RSVD_1):** Reserved
1	0h RW/1C/P	**Unsupported Non-Posted Request Error Status (UNPE):** When set, indicates that P2SB has received unsupported non-posted request;Used only when ENABLE_AER is set
0	0h RW/1C/P	**Unsupported Posted Request Error Status (UPE):** When set, indicates that P2SB has received unsupported posted request;Used only when ENABLE_AER is set

11.47 Unsupported Request Error Control (UREC) — Offset F4h

This register is only reset by a loss of core power

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + F4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Reserved (RSVD):** Reserved
8	0h RO	**Unsupported Request Reporting Enable (URRE):** If set to 1 by software, it allows reporting of an Unsupported Request as a system erro
7:1	0h RO	**Reserved (RSVD_1):** Reserved
0	0h RW/P	**Error Reporting Enable (ERE):** When set, error logged in UES register will trigger error reporting flow in P2SB; Used only when ENABLE_AER is set

11.48 Manufacturers ID (MANID) — Offset F8h

This register is assigned during the boot flow using the SetID message based on values found in the fuse block. All fields except MID will be reset by hardware to zero and set only by the SetID message.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + F8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD):** Reserved
27:24	0h RO/V	**DOT:** Indicates the dot process
23:16	0h RO/V	**Manufacturing Stepping Identifier (MSID):** This field is incremented for each stepping of the part. Note that this field can be used by software to differentiate steppings when the Stepping Revision ID may not change.
15:8	0h RO	**Manufacturing Identifier (MID):** 0Fh = Intel
7:0	0h RO/V	**PROC:** Indicates the process. The dot portion of the process is reflected in bits (27:24)

11.49 SAI Policy Control (SAIPOLCTRL0) — Offset 240h

Bit wise register where each bit represent the 6-bit SAI Index that mapped to a particular 8-bit SAI value or values

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + 240h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**SAI Policy Control 0 (SAIPOLCTRL0):** SAI policy control register bits[31:0]

11.50 SAI Policy Control (SAIPOLCTRL1) — Offset 244h

Bit wise register where each bit represent the 6-bit SAI Index that mapped to a particular 8-bit SAI value or values

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + 244h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**SAI Policy Control 1 (SAIPOLCTRL1):** SAI policy control register bits[63:32]

11.51 SAI Policy Access Control (SAIPOLAC0) — Offset 248h

A value '1 in this register bit means that a Config Cycle with SAI value that match the mapped 8-bit SAI value or values has both the read and write access to the four doublesowrd register for Sideband Message Interface Access

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + 248h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**SAI Policy Access Control 0 (SAIPOLAC0):** SAI policy Access control register bits[31:0]

11.52 SAI Policy Access Control (SAIPOLAC1) — Offset 24Ch

A value '1 in this register bit means that a Config Cycle with SAI value that match the mapped 8-bit SAI value or values has both the read and write access to the four doublesowrd register for Sideband Message Interface Access

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:0] + 24Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**SAI Policy Access Control 1 (SAIPOLAC1):** SAI policy Access control register bits[63:32]

§ §

12 PWM Registers

This chapter documents the registers in Bus: 0, Device 26, Function 0.

12.1 DEVICEVENDORID - Device ID and Vendor ID Register (DEVVENDID) — Offset 0h

Device ID and Vendor ID provided by this register uniquely identifies the particular Device

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + 0h	AC88086 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	AC8h RO/V	**DEVICEVENDORID - Device ID Field (DEVICEID):** Device ID identifies the particular PCI device
15:0	8086h RO/V	**DEVICEVENDORID - Vendor ID Field (VENDORID):** Vendor ID is a unique ID provided by the PCI SIG which identifies the manufacturer of the device

12.2 STATUSCOMMAND — Offset 4h

Command register to programme interrupt disable bus master enable and Memory space enable. Status register to read the errors and aborts

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + 4h	100000 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved Field (RESERVED0):** Reserved
29	0h RW/1C/V	**STATUSCOMMAND- RMA Field (RMA):** Received Master Abort
28	0h RW/1C/V	**STATUSCOMMAND- RTA Field (RTA):** Received Target Abort
27:21	0h RO	**Reserved Field (RESERVED1):** Reserved
20	1h RO	**STATUSCOMMAND- Cap List Field (CAPLIST):** Capabilities List: Indicates that the controller contains a capabilities pointer list
19	0h RO	**STATUSCOMMAND- Intrerrupt Status Field (INTR_STATUS):** Interrupt Status: This bit reflects state of interrupt in the device

Bit Range	Default & Access	Field Name (ID): Description
18:16	0h RO	**Reserved Field (RESERVED2):** Reserved
15:11	0h RO	**Reserved Field (RESERVED3):** Reserved
10	0h RW	**STATUSCOMMAND- Interrupt Disable Field (INTR_DISABLE):** Interrupt Disable
9	0h RO	**Reserved Field (RESERVED4):** Reserved
8	0h RW	**STATUSCOMMAND- SERR Enable Field (SERR_ENABLE):** SERR Enable Not implemented
7:3	0h RO	**Reserved Field (RESERVED5):** Reserved
2	0h RW	**STATUSCOMMAND- BME Field (BME):** Bus Master Enable
1	0h RW	**STATUSCOMMAND- MSE Field (MSE):** Memory Space Enable
0	0h RW/V	**STATUSCOMMAND. IOSR FIELD (IOSE):** Controls a device's response to I/O Space accesses. A value of 0 disables the device response. A value of 1 allows the device to respond to I/O Space accesses. State after RST# is 0. NOTE: This bit does not exist in the PMC IOSF2OCP bridge. It is shadowed in the PSF3 fabric. Using /V in AccessType so PMC cluster validation does not assume this bit will read back what was written.

12.3 REVCLASSCODE — Offset 8h

Revision ID register identifies revision of particular device and Class Code register is used to identify generic function of the device

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + 8h	C800000 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	C8000h RO/V	**REVCLASSCODE - Class code Field (CLASS_CODES):** Class Code register is read-only and is used to identify the generic function of the device and in some cases a specific register-level programming interface
7:0	0h RO/V	**REVCLASSCODE - Revision ID Field (RID):** Revision ID identifies the revision of particular PCI device.

12.4 CLLATHEADERBIST — Offset Ch

Cache Line size as RW with def 0 Latency timer RW with def 0 Header type with Type 0 configuration header and Reserved BIST register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved Field (RESERVED0):** Reserved
23	0h RO	**CLLATHEADERBIST - MultiFunction Device Field (MULFNDEV):** Multi-Function Device
22:16	0h RO	**CLLATHEADERBIST - Header Type Field (HEADERTYPE):** Header Type: Implements Type 0 Configuration header
15:8	0h RO	**CLLATHEADERBIST - Latency Timer Field (LATTIMER):** Latency Timer:. This register is implemented as R/W with default as 0
7:0	0h RW	**CLLATHEADERBIST - Cache Line Size Field (CACHELINE_SIZE):** Cacheline Size

12.5 BAR — Offset 10h

Base Address Register low [31:2] type[2:1] in 32bit or 64bit addr range and memory space indicator [0]

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + 10h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**BAR -Base Address Field (BASEADDR):** Base Address Register Low Base address of the OCP fabric memory space. Taken from Strap values as ones
11:4	0h RO	**BAR -Size Field (SIZEINDICATOR):** Size Indicator RO Always returns 0 The size of this register depends on the size of the memory space
3	0h RO	**BAR -Prefetchable Field (PREFETCHABLE):** Prefetchable: Indicates that this BAR is not prefetchable
2:1	0h RO	**BAR -Type Field (TYPE):** If BAR_64b_EN is 0 then 00 indicates BAR lies in 32bit address range If BAR_64b_EN is 1 then 10 Indicates BAR lies in 64 bit address range
0	0h RO	**BAR - Message Space Field (MESSAGE_SPACE):** Memory Space Indicator: 0 indicates this BAR is present in the memory space.

12.6 BAR -Base Address Register High (BAR_HIGH) — Offset 14h

Base Address Register High enabled if [2:1] of BAR_type_LOW is 10

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + 14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**BAR -Base Address High Field (BASEADDR_HIGH):** Base Address high - MSB

12.7 BAR1 — Offset 18h

Base Address Register1 accesses to PCI configuration space and is always 4K type in [2:1] and memory space indicator in [0]

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + 18h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**BAR1 - Base Address Field (BASEADDR1):** Base Address1 This field is present if BAR1 is enabled through private configuration space.
11:4	0h RO	**BAR1 -Size Field (SIZEINDICATOR1):** Always is 0 as minimum size is 4K
3	0h RO	**BAR1 - Prefetchable Field (PREFETCHABLE1):** Prefetchable: Indicates that this BAR is not prefetchable.
2:1	0h RO	**BAR1 -Type Field (TYPE1):** If BAR_64b_EN is 0 then 00 indicates BAR lies in 32bit address range If BAR_64b_EN is 1 then 10 Indicates BAR lies in 64 bit address range
0	0h RO	**BAR1 - Message Space Field (MESSAGE_SPACE1):** Memory Space Indicator: 0 Indicates this BAR is present in the memory space

12.8 BAR1 -Base Address Register1 High (BAR1_HIGH) — Offset 1Ch

Base Address Register1 High enabled only if [2:1] of BAR1 register is 10

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + 1Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**BAR1 -Base Address High Field (BASEADDR1_HIGH):** Base Address: Base address of the OCP fabric memory space. Taken from Strap values as ones

12.9 BAR2 — Offset 20h

BAR -Base Address Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + 20h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**BASEADDR:**
1	0h RO	**RESERVED0:** Reserved
0	0h RO	**MESSAGE_SPACE:**

12.10 SUBSYSTEMID — Offset 2Ch

SVID register along with SID register is to distinguish subsystem from another

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**SUBSYSTEMID:** Subsystem ID: This register is implemented for any function that can be instantiated more than once in a given system.
15:0	0h RW/O	**SUBSYSTEMID -Subsystem Vendor Field (SUBSYSTEMVENDORID):** Subsystem Vendor ID: This register must be implemented for any function that can be instantiated more than once in a given system

12.11 EXPANSION ROM base address (EXPANSION_ROM_BASEADDR) — Offset 30h

EXPANSION ROM base address register is a RO indicates support for expansion ROMs

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + 30h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**EXPANSION ROM base address field (EXPANSION_ROM_BASE):** Expansion ROM Base Address: Value of all zeros indicates no support for Expansion ROM

12.12 CAPABILITYPTR — Offset 34h

Capabilities Pointer register indicates what the next capability is

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + 34h	80 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved Field (RESERVED0):** Reserved
7:0	80h RO	**CAPABILITYPTR - Capabilities Pointer Field (CAPPTR_POWER):** Capabilities Pointer: Indicates what the next capability is.

12.13 INTERRUPTREG — Offset 3Ch

Interrupt line Register isn't used in Bridge directly Interrupt Pin register reflects the IPIN value in private config space. Min_gnt register indicating the req of latency timers and max_lat register max latenc

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + 3Ch	100 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**INTERRUPTREG - Max Latency Field (MAX_LAT):** Value of 0 indicates device has no major requirements for the settings of latency timers
23:16	0h RO	**INTERRUPTREG - Min Gnt Field (MIN_GNT):** Value of 0 indicates device has no major requirements for the settings of latency timers // Re-Check Desc Padma
15:12	0h RO	**Reserved Field (RESERVED0):** Reserved
11:8	1h RO	**INTERRUPTREG - Interrupt Pin Field (INTPIN):** Interrupt Pin: Value in this register is reflected from the IPIN value in the private configuration space.
7:0	0h RW	**INTERRUPTREG - Int Line Field (INTLINE):** Interrupt Line: It is used to communicate to software the interrupt line to which the interrupt pin is connected

12.14 POWERCAPID — Offset 80h

PowerManagement Capability ID register points to next capability structure and power mgmnt capability with Power management capabilities register for PME support and version

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + 80h	48030001 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	9h RO	**POWERCAPID - PME Support Field (PMESUPPORT):** This 5-bit field indicates the power states in which the function can assert the PME#
26:19	0h RO	**Reserved Field (RESERVED0):** Reserved
18:16	3h RO	**POWERCAPID - Version Field (VERSION):** Indicates support for Revision 1.2 of the PCI Power Management Specification
15:8	0h RO	**POWERCAPID - Next Cap Field (NXTCAP):** Next Capability: Points to the next capability structure.
7:0	1h RO	**POWERCAPID - Power Capability ID Field (POWER_CAP):** Power Management Capability: Indicates this is power management capability

12.15 PMECTRLSTATUS — Offset 84h

power management control and status register to set and read PME status PME enable No Soft reset and power state

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + 84h	8 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved Field (RESERVED0):** Reserved
15	0h RW/1C/V	**PMECTRLSTATUS - PME Status Field (PMESTATUS):** PME Status
14:9	0h RO	**Reserved Field (RESERVED1):** Reserved
8	0h RW	**PMECTRLSTATUS - PME Enable Field (PMEENABLE):** PME Enable
7:4	0h RO	**Reserved Field (RESERVED2):** Reserved
3	1h RO	**PMECTRLSTATUS - No Soft Reset Field (NO_SOFT_RESET):** This bit indicates that devices transitioning from D3hot to D0 because of Powerstate commands do not perform an internal reset
2	0h RO	**Reserved Field (RESERVED3):** Reserved
1:0	0h RW	**PMECTRLSTATUS - Power State Field (POWERSTATE):** Power State: This field is used both to determine the current power state and to set a new power state

12.16 PCI DEVICE IDLE CAPABILITY RECORD (PCIDEVIDLE_CAP_RECORD) — Offset 90h

PCI Device Vendor Specific Capability register defines Vendor specific Capability ID, revision , length , next capability and CAPID

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + 90h	F0140009 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RO	**PCIDEVIDLE_CAP_REG - Vendor Cap Field (VEND_CAP):** Vendor Specific Capability ID
27:24	0h RO	**PCIDEVIDLE_CAP_REG - Revision IDField (REVID):** Revision ID of capability structure
23:16	14h RO	**PCIDEVIDLE_CAP_REG - Cap Length Field (CAP_LENGTH):** Vendor Specific Capability Length
15:8	0h RO	**PCIDEVIDLE_CAP_REG - Next Capability Field (NEXT_CAP):** Next Capability
7:0	9h RO	**PCIDEVIDLE_CAP_REG - Capability ID Field (CAPID):** Capability ID

12.17 DEVID VENDOR SPECIFIC REG (DEVID_VEND_SPECIFIC_REG) — Offset 94h

Extended Vendor capability register for VSEC Length revision and ID

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + 94h	1400010 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	14h RO	**DEVID VENDOR SPECIFIC REG - Vendor Specific ID Field (VSEC_LENGTH):** Vendor Specific Extended Capability Length
19:16	0h RO	**DEVID VENDOR SPECIFIC REG - Vendor Specific Revision Field (VSEC_REV):** Vendor specific Extended Capability revision
15:0	10h RO	**DEVID VENDOR SPECIFIC REG - Vendor Specific Length Field (VSECID):** Vendor Specific Extended Capability ID

12.18 D0I3_CONTROL_SW_LTR_MMIO_REG — Offset 98h

Software location pointer in MMIO space as an offset specified by BAR

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + 98h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**D0I3_CONTROL_SW_LTR_MMIO_REG - SW LTR Dword Offset Field (SW_LAT_DWORD_OFFSET):** SW LTR Update MMIO Offset Location (SWLTRLOC)
3:1	0h RO	**D0I3_CONTROL_SW_LTR_MMIO_REG - SW LTR BAR Number Field (SW_LAT_BAR_NUM):** Indicates that the SW LTR update MMIO location is always at BAR0
0	0h RO	**D0I3_CONTROL_SW_LTR_MMIO_REG - SW LTR Valid Field (SW_LAT_VALID):** This value is reflected from the SW LTR valid strap at the top level

12.19 DEVICE_IDLE_POINTER_REG — Offset 9Ch

Device IDLE pointer register giving details on Device MMIO offset location BAR NUM and D0i3 Valid Strap

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + 9Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**DEVICE_IDLE_POINTER_REG - D0i3 Dword Offset Field (DWORD_OFFSET):** contains the location pointer to the SW LTR register in MMIO space as an offset from the specified BAR
3:1	0h RO	**DEVICE_IDLE_POINTER_REG - BAR NUM Field (BAR_NUM):** Bar num: Indicates that the D0i3 MMIO location is always at BAR0
0	0h RO	**DEVICE_IDLE_POINTER_REG - D0i3 Valid Field (VALID):** Valid: This value is reflected from the D0i3 valid strap at the top level.

12.20 D0I3_MAX_POW_LAT_PG_CONFIG — Offset A0h

D0idle_Max_Power_On_Latency register set at boot and Power control enable register to enable communication with the PGCB block below the Bridge

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + A0h	800 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved Field (RESERVED0):** Reserved
21	0h RW	**D0I3_MAX_POW_LAT_PG_CONFIG - HAE Field (HAE):** Hardware Autonomous Enable
20	0h RO	**Reserved Field (RESERVED1):** Reserved
19	0h RW	**D0I3_MAX_POW_LAT_PG_CONFIG - Sleep Enable Field (SLEEP_EN):** Sleep Enable

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW	**D0I3_MAX_POW_LAT_PG_CONFIG - D3 Hen Field (PGE):** DEVIDLE Enable (DEVIDLEN): If ?1? then the function will power gate when idle and the DevIdle register (DevIdleC[2] = ?1?) is set.
17	0h RW	**D0I3_MAX_POW_LAT_PG_CONFIG - Device Idle En Field (I3_ENABLE):** D3-Hot Enable (D3HEN): If ?1?, then function will power gate when idle and the PMCSR[1:0] register in the function =?11? (D3).
16	0h RW	**D0I3_MAX_POW_LAT_PG_CONFIG - PMC Request Enable Field (PMCRE):** PMCRE: PMC Request Enable
15:13	0h RO	**Reserved Field (RESERVED2):** Reserved
12:10	2h RW/O	**D0I3_MAX_POW_LAT_PG_CONFIG - Power Latency Scale Field (POW_LAT_SCALE):** Power On Latency Scale
9:0	0h RW/O	**D0I3_MAX_POW_LAT_PG_CONFIG - Power Latency Value Field (POW_LAT_VALUE):** Power On Latency value

12.21 GEN_REGRW1 - General Purpose Read Write Register1 (GEN_REGRW1) — Offset B0h

General Purpose PCI Read Write Register1

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + B0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REGRW1 - General Purpose Read Write Field (GEN_REG_RW1):** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg1 Out of Band signal

12.22 GEN_REGRW2 — Offset B4h

General Purpose PCI Read Write Register2

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + B4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REGRW2 - General Purpose Read Write Field (GEN_REG_RW2):** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg2 Out of Band signal

12.23 GEN_REGRW3 — Offset B8h

General Purpose PCI Read Write Register3

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + B8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REGRW3 - General Purpose Read Write Field (GEN_REG_RW3):** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg3 Out of Band signal

12.24 GEN_REGRW4 — Offset BCh

General Purpose PCI Read Write Register4

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + BCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REGRW4 - General Purpose Read Write Field (GEN_REG_RW4):** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg4 Out of Band signal

12.25 GEN_INPUT_REG — Offset C0h

General Purpose Input Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + C0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**GEN_INPUT_REG - General Purpose Input Field (GEN_REG_INPUT_RW):** General Purpose Input Register: This register value reflects the value of oob_gen_input_pci Out of Band signal

12.26 Manufacturers ID (MANID) — Offset F8h

MAN ID Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:26, F:0] + F8h	4000F1C h

Bit Range	Default & Access	Field Name (ID): Description
31:0	4000F1Ch RO/V	**Manufacturers ID - MAN ID (MANID):** Manufacturer ID: Default value comes from straps.

§ §

13 SD and MMC Registers

This chapter documents the registers of the LPIO controller devices. The processor contains multiple PCIe controller devices:

- Bus: 0, Device: 28, Function: 0 (SDMMC0)
- Bus: 0, Device: 30, Function: 0 (SDMMC1)
- Bus: 0, Device: 27, Function: 0 (SDMMC2)

NOTE: Register default values are taken from device SDMMC0 only. Refer Vol1 for Device IDs.

13.1 DEVVENDID — Offset 0h

DEVICEVENDORID - Device ID and Vendor ID Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + 0h	8086 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**DEVICEID:**
15:0	8086h RO	**VENDORID:**

13.2 STATUSCOMMAND — Offset 4h

STATUSCOMMAND- Status and Command

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + 4h	100000 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**RESERVED0:** Reserved
29	0h RW/1C	**RMA:**
28	0h RW/1C	**RTA:**
27:21	0h RO	**RESERVED1:** Reserved

Bit Range	Default & Access	Field Name (ID): Description
20	1h RO	**CAPLIST:**
19	0h RO	**INTR_STATUS:**
18:16	0h RO	**RESERVED2:** Reserved
15:11	0h RO	**RESERVED3:** Reserved
10	0h RW	**INTR_DISABLE:**
9	0h RO	**RESERVED4:** Reserved
8	0h RW	**SERR_ENABLE:**
7:3	0h RO	**RESERVED5:** Reserved
2	0h RW	**BME:**
1	0h RW	**MSE:**
0	0h RO	**RESERVED6:** Reserved

13.3 REVCLASSCODE — Offset 8h

REVCLASSCODE - Revision ID and Class Code

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + 8h	8050100 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	80501h RO	**CLASS_CODES:**
7:0	0h RO	**RID:**

13.4 CLLATHEADERBIST — Offset Ch

CLLATHEADERBIST - Cache Line Latency Header and BIST

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**RESERVED0:** Reserved
23	0h RO	**MULFNDEV:**
22:16	0h RO	**HEADERTYPE:**
15:8	0h RO	**LATTIMER:**
7:0	0h RW	**CACHELINE_SIZE:**

13.5 BAR — Offset 10h

BAR -Base Address Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + 10h	4 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**BASEADDR:**
11:4	0h RO	**SIZEINDICATOR:**
3	0h RO	**PREFETCHABLE:**
2:1	2h RO	**TYPE:**
0	0h RO	**MESSAGE_SPACE:**

13.6 BAR_HIGH — Offset 14h

BAR -Base Address Register High

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + 14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**BASEADDR_HIGH:**

13.7 BAR1 — Offset 18h

BAR1 -Base Address Register1

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + 18h	4 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**BASEADDR1:**
11:4	0h RO	**SIZEINDICATOR1:**
3	0h RO	**PREFETCHABLE1:**
2:1	2h RO	**TYPE1:**
0	0h RO	**MESSAGE_SPACE1:**

13.8 BAR1_HIGH — Offset 1Ch

BAR1 -Base Address Register1 High

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + 1Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**BASEADDR1_HIGH:**

13.9 SUBSYSTEMID — Offset 2Ch

SUBSYSTEMID -Subsystem Vendor and Subsystem ID

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**SUBSYSTEMID:**

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW/O	**SUBSYSTEMVENDORID:**

13.10 EXPANSION_ROM_BASEADDR — Offset 30h

EXPANSION ROM base address

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + 30h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**EXPANSION_ROM_BASE:**

13.11 CAPABILITYPTR — Offset 34h

CAPABILITYPTR - Capabilities Pointer

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + 34h	80 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**RESERVED0:** Reserved
7:0	80h RO	**CAPPTR_POWER:**

13.12 INTERRUPTREG — Offset 3Ch

INTERRUPTREG - Interrupt Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + 3Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**MAX_LAT:**

Bit Range	Default & Access	Field Name (ID): Description
23:16	0h RO	**MIN_GNT:**
15:12	0h RO	**RESERVED0:** Reserved
11:8	0h RO	**INTPIN:**
7:0	0h RW	**INTLINE:**

13.13 POWERCAPID — Offset 80h

POWERCAPID - PowerManagement Capability ID

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + 80h	39001 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	0h RO	**PMESUPPORT:**
26:19	0h RO	**RESERVED0:** Reserved
18:16	3h RO	**VERSION:**
15:8	90h RO	**NXTCAP:**
7:0	1h RO	**POWER_CAP:**

13.14 PMECTRLSTATUS — Offset 84h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + 84h	8 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED0:** Reserved
15	0h RW/1C	**PMESTATUS:**
14:9	0h RO	**RESERVED1:** Reserved

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW	**PMEENABLE:**
7:4	0h RO	**RESERVED2:** Reserved
3	1h RO	**NO_SOFT_RESET:**
2	0h RO	**RESERVED3:** Reserved
1:0	0h RW	**POWERSTATE:**

13.15 PCIDEVIDLE_CAP_RECORD — Offset 90h

PCI DEVICE IDLE CAPABILITY RECORD

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + 90h	F0140009 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RO	**VEND_CAP:**
27:24	0h RO	**REVID:**
23:16	14h RO	**CAP_LENGTH:**
15:8	0h RO	**NEXT_CAP:**
7:0	9h RO	**CAPID:**

13.16 DEVID_VEND_SPECIFIC_REG — Offset 94h

DEVID VENDOR SPECIFIC REG

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + 94h	1400010 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	14h RO	**VSEC_LENGTH:**

Bit Range	Default & Access	Field Name (ID): Description
19:16	0h RO	**VSEC_REV:**
15:0	10h RO	**VSECID:**

13.17 D0I3_CONTROL_SW_LTR_MMIO_REG — Offset 98h

SW LTR Update MMIO Location Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + 98h	8041 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	804h RO	**SW_LAT_DWORD_OFFSET:**
3:1	0h RO	**SW_LAT_BAR_NUM:**
0	1h RO	**SW_LAT_VALID:**

13.18 DEVICE_IDLE_POINTER_REG — Offset 9Ch

Device IDLE pointer register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + 9Ch	81C1 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	81Ch RO	**DWORD_OFFSET:**
3:1	0h RO	**BAR_NUM:**
0	1h RO	**VALID:**

13.19 D0I3_MAX_POW_LAT_PG_CONFIG — Offset A0h

DEVICE PG CONFIG

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + A0h	290800 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**RESERVED0:** Reserved
21	1h RW	**HAE:**
20	0h RO	**RESERVED1:** Reserved
19	1h RW	**SLEEP_EN:**
18	0h RW	**PGE:**
17	0h RW	**I3_ENABLE:**
16	1h RW	**PMCRE:**
15:13	0h RO	**RESERVED2:** Reserved
12:10	2h RW/O	**POW_LAT_SCALE:**
9:0	0h RW/O	**POW_LAT_VALUE:**

13.20 GEN_REGRW1 — Offset B0h

General Purpose Read Write Register1

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + B0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW1:**

13.21 GEN_REGRW2 — Offset B4h

General Purpose Read Write Register2

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + B4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW2:**

13.22 GEN_REGRW3 — Offset B8h

General Purpose Read Write Register3

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + B8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW3:**

13.23 GEN_REGRW4 — Offset BCh

General Purpose Read Write Register4

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + BCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW4:**

13.24 GEN_INPUT_REG — Offset C0h

General Purpose Input Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + C0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**GEN_REG_INPUT_RW:**

13.25 MANID — Offset F8h

Manufacturers ID

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:27, F:0] + F8h	F h

Bit Range	Default & Access	Field Name (ID): Description
31:0	Fh RO	**MANID**

13.26 SW_LTR_Value (SW_LTR_val)—Offset 804h

Sets the Software Latency Tolerance Reporting fields

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 800h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	Reserved.
15	0h RW	**Snoop_Requirment (Snoop_Requirment):** If the Requirement (bit 15) is clear, that indicates that the device has no LTR requirement for this type of traffic (i.e. it can wait for service indefinitely). If the 10-bit latency value is zero it indicates that the device cannot tolerate any delay and needs the best possible service/response time.
14:13	0h RO	Reserved.
12:10	2h RW	**Snoop_latency_scale (Snoop_latency_scale):** Support for codes 010 (1us) or 011 (32us) for Snoop Latency Scale(1us -) 32ms total span) only. Writes to this CSR which dont match those values will be dropped completely, next read will return previous value.
9:0	0h RW	**Snoop_value (Snoop_value):** 10-bit latency value

13.27 Auto LTR Value (Auto_LTR_val)—Offset 808h

Sets the Auto Latency Tolerance Reporting fields

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 800h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	Reserved.
15	0h RW	**Snoop_Requirment (Snoop_Requirment):** If the Requirement (bit 15) is clear, that indicates that the device has no LTR requirement for this type of traffic (i.e. it can wait for service indefinitely). If the 10-bit latency value is zero it indicates that the device cannot tolerate any delay and needs the best possible service/response time.
14:13	0h RO	**Reserved_low field (Reserved_low):** Reserved_low
12:10	2h RW	**Snoop_latency_scale (Snoop_latency_scale):** Support for codes 010 (1us) or 011 (32us) for Snoop Latency Scale(1us -) 32ms total span) only. Writes to this CSR which dont match those values will be dropped completely, next read will return previous value.
9:0	0h RW	**Snoop_value (Snoop_value):** 10-bit latency value

13.28 Capabilities Bypass Control (Cap_byps)—Offset 810h

Capabilities Bypass Control register

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7:0	0h RW	**Enable Cap Bypass (Enable_Cap_Bypass):** 8h5A Enable Capabilities Bypass. All other Capabilities Bypass Disable (using default values)

13.29 emmc Capabilities Bypass Register I (Cap_byps_reg1)—Offset 814h

eMMC Capabilities Bypass register 1

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 3041EF3Ch

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	Reserved.
29	1h RW	**hs400 support (hs400_support):** 1: HS400 Mode Supported. 0: HS400 Mode NOT Supported
28	1h RW	**timeout clock unit (timeout_clock_unit):** 1b1 - to Select MHz Clock ,1b0 - to Select KHz Clock
27:22	1h RW	**timeout clock freq (timeout_clock_freq):** Timeout clock frequency
21	0h RW	**SPI mode support (SPI_mode_support):** SPI Mode Support 1b1 SPI Mode Supported ,1b0 SPI Mode Not Supported
20:17	0h RW	**timer count (timer_count):** Timer Count for Re-Tuning This is the Timer Count for Re-Tuning Timer for Re-Tuning Mode 1 to 3. Setting to 4b0 disables Re-Tuning Timer
16	1h RW	**tuning for SDR50 (tuning_for_SDR50):** Use Tuning for SDR50 1b1 Use Tuning 1b0 Dont use Tuning
15	1h RW	**ddr50 support (ddr50_support):** 1: DDR50 Mode Supported. 0: DDR50 Mode NOT Supported
14	1h RW	**sdr104 support (sdr104_support):** 1: SDR104 Mode Supported. 0: SDR104 Mode NOT Supported
13	1h RW	**sdr50 support (sdr50_support):** 1: SDR50 Mode Supported. 0: SDR50 Mode NOT Supported
12:11	1h RW	**Slot Type (Slot_Type):** 00 - Removable SD Card Slot. 01 - Embedded Slot for One Device. 10 - Shared Bus Slot. 11 - Reserved
10	1h RW	**Async Interrupt Support (Async_Interrupt_Support):** 1: Asynchronous Interrupt Supported. 0: Asynchronous Interrupt NOT Supported
9	1h RW	**Sys Addr 64bit Support (Sys_Addr_64bit_Support):** 1 - Core supports 64-bit System Address Bus. 0 - Core supports only 32-bit System Address Bus
8	1h RW	**Voltage Support 1_8V (Voltage_Support_1_8V):** 1: 1.8V Supported. 0: 1.8V NOT Supported
7	0h RW	**Voltage Support 3V (Voltage_Support_3V):** 1: 3.0V Supported. 0: 3.0V NOT Supported

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**Voltage Support 3_3V (Voltage_Support_3_3V):** 1: 3.3V Supported. 0: 3.3V NOT Supported
5	1h RW	**Suspend Resume Support (Suspend_Resume_Support):** 1: Suspend/Resume Supported. 0: Suspend/Resume NOT Supported
4	1h RW	**SDMA Support (SDMA_Support):** 1: SDMA mode Supported. 0: SDMA mode NOT Supported
3	1h RW	**High Speed Support (High_Speed_Support):** 1: HIGH_SPEED mode Supported. 0: HIGH_SPEED mode NOT Supported
2	1h RW	**ADMA2 Support (ADMA2_Support):** 1: ADMA2 mode Supported. 0: ADMA2 mode NOT Supported
1:0	0h RW	**Max Burst Length (Max_Burst_Length):** Maximum Block Length supported by the Core/Device. 00: 512 (Bytes). 01: 1024. 10: 2048. 11: Reserved

13.30 emmc Capabilities Bypass Register 2 (Cap_byps_reg2)—Offset 818h

eMMC Capabilities Bypass register 2

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 40040C8h

Bit Range	Default & Access	Field Name (ID): Description
31:27	0h RO	Reserved.
26:21	20h RW	**tuning count val (tuning_count_val):** Tuning Count Value Configures the Number of Taps (Phases) of the rxclk_in that is supported. The Tuning State machine uses this information to select one of the Taps (Phases) of the rxclk_in during the Tuning Procedure.
20	0h RW	**tuning dis (tuning_dis):** Disable the 1.5x Tuning count when calculating total tuning count. The internal tuning count will be set to the corecfg_Tuningcount when this signal is asserted
19	0h RW	**driver type 4 (driver_type_4):** Driver Type 4 Support 1b1 Supported 1b0 NOT Supported
18	0h RW	**driver type D (driver_type_D):** Driver Type D Support 1b1 Supported 1b0 NOT Supported
17	0h RW	**driver type C (driver_type_C):** Driver Type C Support 1b1 Supported 1b0 NOT Supported
16	0h RW	**driver type A (driver_type_A):** Driver Type A Support 1b1 Supported 1b0 NOT Supported

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	Reserved.
14	1h RW	**support 8-bit embedded (support_8_bit_embedded):** 8-bit Support for Embedded Device 1b1 Supported 1b0 NOT Supported
13:8	0h RO	Reserved.
7:0	C8h RW	**base sd clock (base_sd_clock):** Base Clock Frequency for SD Clock

13.31 D0i3 (reg_D0i3)—Offset 81Ch

D0i3 Control register

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 8h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	Reserved.
3	1h RW/1C	**Restore Required (RestoreRequired):** When set (by HW), SW must restore state to the IP. The state may have been lost due to a reset or full power lost. SW clears the bit by writing a 1. This bit will be set on initial power up.
2	0h RW	**D0i3 (D0i3):** SW sets this bit to 1 to move the IP into the D0i3 state. Writing this bit to 0 will return the IP to the fully active D0 state (D0i0).
1	0h RO	**Interrupt Request (Interrupt_Request):** SW sets this bit to 1 to ask for an interrupt to be generated on completion of the command. SW must clear or set this on each write to this register. Not supported in SCS!
0	0h RO	**Cmd In Progress (Cmd_In_Progress):** HW sets this bit on a 1->0 or 0->1 transition of bit [2]. While set, the other bits in this register are not valid and it is illegal for SW to write to any bit in this register. When clear all the other bits in the register are valid and SW may write to any bit. If Interrupt Request bit [1] was set for the current command, HW may clear this bit before the interrupt has been made visible to SW, since when SW actually handles

13.32 Tx CMD Delay Control (Tx_CMD_dly)—Offset 820h

Front end module Tx Command Path Delay register

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 500h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	Reserved.
14:8	5h RW	**ddr mode (ddr_mode):** Tx CMD Delay (DDR Mode). 0 39 Select number of active delay elements. Each = 125pSec. 40 127 Reserved
7	0h RO	Reserved.
6:0	0h RW	**sdr mode (sdr_mode):** Tx CMD Delay (SDR Mode). 0 39 Select number of active delay elements. Each = 125pSec. 40 127 - Reserved

13.33 Tx DATA Delay Control 1 (Tx_DATA_dly_1)—Offset 824h

Front end module Tx data Path Delay register 1

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: A18h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	Reserved.
14:8	Ah RW	**hs400 mode (hs400_mode):** Tx Data Delay (HS400 Mode). 0 78 Select number of active delay elements. Each = 125pSec. 79 127 - Reserved
7	0h RO	Reserved.
6:0	18h RW	**sdr104 hs200 (sdr104_hs200):** Tx Data Delay (SDR104/HS200 Mode) 0-79 - Select the required delay, as a multiple of 125pSec.80 127 Reserved

13.34 Tx DATA Delay Control 2 (Tx_DATA_dly_2)—Offset 828h

Front end module Tx data Path Delay register 2

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 1C2A1C00h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	Reserved.
30:24	1Ch RW	**sdr50 mode (sdr50_mode):** Tx Data Delay (SDR50 Mode) 0-79 - Select the required delay, as a multiple of 125pSec. 80 127 Reserved
23	0h RO	Reserved.
22:16	2Ah RW	**ddr50 mode (ddr50_mode):** Tx Data Delay (DDR50 Mode). 0 78 Select number of active delay elements. Each = 125pSec. 79 127 - Reserved
15	0h RO	Reserved.
14:8	1Ch RW	**sdr25 hs50 mode (sdr25_hs50_mode):** Tx Data Delay (SDR25/ HS50 Mode)0-79 - Select the required delay, as a multiple of 125pSec.80 127 Reserved
7	0h RO	Reserved.
6:0	0h RW	**sdr12 comp mode (sdr12_comp_mode):** Tx Data Delay (SDR12/Compatibility Mode) 0-79 - Select the required delay, as a multiple of 125pSec.80 127 Reserved

13.35 Rx CMD Data Delay Control 1 (Rx_CMD_Data_dly_1)—Offset 82Ch

Front end module Rx data Path Delay register 1

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: E0000h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	Reserved.
30:24	0h RW	**sdr50 mode (sdr50_mode):** Rx CMD + Data Delay (SDR50 Mode). 0 79 Select the required delay, as a multiple of 125pSec.80 127 Reserved
23	0h RO	Reserved.

Bit Range	Default & Access	Field Name (ID): Description
22:16	Eh RW	**ddr50 mode (ddr50_mode):** Rx CMD + Data Delay (DDR50 Mode). 0 78 Select number of active delay elements. Each = 125pSec. 79 127 - Reserved
15	0h RO	Reserved.
14:8	0h RW	**sdr25 hs50 (sdr25_hs50):** Rx CMD + Data Delay (SDR25/HS50 Mode) 0-79 - Select the required delay, as a multiple of 125pSec.80 127 Reserved
7	0h RO	Reserved.
6:0	0h RW	**sdr12 comp (sdr12_comp):** Rx CMD + Data Delay (SDR12/ Compatibility Mode) 0-79 - Select the required delay, as a multiple of 125pSec.80 127 Reserved

13.36 Rx Strobe Delay Control (Rx_Strobe_Ctrl_Path)—Offset 830h

Front end module Rx strobe Path Delay register

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	Reserved.
16	0h RW	**auto tuning (auto_tuning):** Enable Auto Tuning for HS400 Strobe Path 0 Auto Tuning Disabled 1 Auto Tuning Enabled
15	0h RO	Reserved.
14:8	0h RW	**hs400 mode1 (hs400_mode1):** Rx Strobe Delay DLL 1(HS400 Mode) 0 39 Select number of active delay elements. Each = 125pSec 0 63 - Reserved
7	0h RO	Reserved.
6:0	0h RW	**hs400 mode2 (hs400_mode2):** Rx Strobe Delay DLL 2(HS400 Mode) 0 39 Select number of active delay elements. Each = 125pSec 40 63 - Reserved

13.37 Rx CMD Data Delay Control 2 (Rx_CMD_Data_dly_2)—Offset 834h

Front end module Rx data Path Delay register 2

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 10000h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	Reserved.
17:16	1h RW	**clk source (clk_source):** Clock Source for Rx Path 00 Rx Clock after Output Buffer 01 Rx Clock before Output Buffer 10 Automatic Selection based on Working mode (HS 200 before buffer, all others after buffer) 11 - Reserved
15:14	0h RO	Reserved.
13:8	0h RW	**path pll (path_pll):** Rx Path PLL #3 Delay value For Auto Tuning Mode 0-39 - Select the required delay, as a multiple of 125pSec.40 63 Reserved
7	0h RO	Reserved.
6:0	0h RW	**cmd data sdr104 hs200 (cmd_data_sdr104_hs200):** Rx CMD + Data Delay (SDR104/HS200 Mode) 0-79 - Select the required delay, as a multiple of 125pSec. 80 127 Reserved

13.38 Master DLL Software Control (Master_Dll)—Offset 838h

Master DLL Software control register

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	Reserved.
24	0h RW	**SW reset dll (SW_reset_dll):** SW reset for Master DLL 0 No SW Reset for Master DLL 1 Force Reset for Master DLL
23	0h RO	**DLL lock (DLL_lock):** Master DLL Lock Indication
22:8	0h RW	**coarse code (coarse_code):** Set coarse code to DLL. (only valid when Software control is Enabled)
7:4	0h RW	**fine code (fine_code):** Set fine code to DLL. (only valid when Software control is Enabled)

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO/V	**less (less):** Phase Detection Less Indication
2	0h RO/V	**more (more):** Phase Detector More Indication
1	0h RW	**Master DLL Software Ctrl (Master_DLL_Software_Ctrl):** Master DLL Software Ctrl. 0 Master DLL Automatic Control (SW Control Disabled). 1 Master DLL Software Control Enabled
0	1h RW	**Ctrl of Mst DLL Ref Clk (Ctrl_of_Mst_DLL_Ref_Clk):** Ctrl of Master DLL Ref Clock. 0 - Clock is Disabled. 1 - Clock is Enabled

13.39 Auto_tuning_val (Auto_tuning)—Offset 840h

Auto Tuning Value

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RO	Reserved.
4:0	0h RO/V	**Auto_tuning_val (Auto_tuning_val):** Auto tuning value

13.40 Root Space sel (emmc_Root_Space)—Offset 900h

Root space select to indicate which root space the IP will output its upstream accesses

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	Reserved.
0	0h RO	**root space (root_space):** Root Space Select Selects the Root Space in which the IP output its upstream accesses 0 Root Space 0 1 Root Space 1

13.41 SW_LTR_Value (SW_LTR_val)—Offset 804h

Sets the Software Latency Tolerance Reporting fields

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 800h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	Reserved.
15	0h RW	**Snoop_Requirment (Snoop_Requirment):** If the Requirement (bit 15) is clear, that indicates that the device has no LTR requirement for this type of traffic (i.e. it can wait for service indefinitely). If the 10-bit latency value is zero it indicates that the device cannot tolerate any delay and needs the best possible service/response time.
14:13	0h RO	Reserved.
12:10	2h RW	**Snoop_latency_scale (Snoop_latency_scale):** Support for codes 010 (1us) or 011 (32us) for Snoop Latency Scale(1us -) 32ms total span) only. Writes to this CSR which dont match those values will be dropped completely, next read will return previous value.
9:0	0h RW	**Snoop_value (Snoop_value):** 10-bit latency value

13.42 Auto LTR Value (Auto_LTR_val)—Offset 808h

Sets the Auto Latency Tolerance Reporting fields

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 800h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	Reserved.
15	0h RW	**Snoop_Requirment (Snoop_Requirment):** If the Requirement (bit 15) is clear, that indicates that the device has no LTR requirement for this type of traffic (i.e. it can wait for service indefinitely). If the 10-bit latency value is zero it indicates that the device cannot tolerate any delay and needs the best possible service/response time.

Bit Range	Default & Access	Field Name (ID): Description
14:13	0h RO	**Reserved_low field (Reserved_low):** Reserved_low
12:10	2h RW	**Snoop_latency_scale (Snoop_latency_scale):** Support for codes 010 (1us) or 011 (32us) for Snoop Latency Scale(1us -) 32ms total span) only. Writes to this CSR which dont match those values will be dropped completely, next read will return previous value.
9:0	0h RW	**Snoop_value (Snoop_value):** 10-bit latency value

13.43 Capabilities Bypass Control (Cap_byps)—Offset 810h

Capabilities Bypass Control register

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7:0	0h RW	**Enable Cap Bypass (Enable_Cap_Bypass):** 8h5A Enable Capabilities Bypass. All other Capabilities Bypass Disable (using default values)

13.44 emmc Capabilities Bypass Register I (Cap_byps_reg1)—Offset 814h

eMMC Capabilities Bypass register 1

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 3041EF3Ch

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	Reserved.
29	1h RW	**hs400 support (hs400_support):** 1: HS400 Mode Supported. 0: HS400 Mode NOT Supported

Bit Range	Default & Access	Field Name (ID): Description
28	1h RW	**timeout clock unit (timeout_clock_unit):** 1b1 - to Select MHz Clock ,1b0 - to Select KHz Clock
27:22	1h RW	**timeout clock freq (timeout_clock_freq):** Timeout clock frequency
21	0h RW	**SPI mode support (SPI_mode_support):** SPI Mode Support 1b1 SPI Mode Supported ,1b0 SPI Mode Not Supported
20:17	0h RW	**timer count (timer_count):** Timer Count for Re-Tuning This is the Timer Count for Re-Tuning Timer for Re-Tuning Mode 1 to 3. Setting to 4b0 disables Re-Tuning Timer
16	1h RW	**tuning for SDR50 (tuning_for_SDR50):** Use Tuning for SDR50 1b1 Use Tuning 1b0 Dont use Tuning
15	1h RW	**ddr50 support (ddr50_support):** 1: DDR50 Mode Supported. 0: DDR50 Mode NOT Supported
14	1h RW	**sdr104 support (sdr104_support):** 1: SDR104 Mode Supported. 0: SDR104 Mode NOT Supported
13	1h RW	**sdr50 support (sdr50_support):** 1: SDR50 Mode Supported. 0: SDR50 Mode NOT Supported
12:11	1h RW	**Slot Type (Slot_Type):** 00 - Removable SD Card Slot. 01 - Embedded Slot for One Device. 10 - Shared Bus Slot. 11 - Reserved
10	1h RW	**Async Interrupt Support (Async_Interrupt_Support):** 1: Asynchronous Interrupt Supported. 0: Asynchronous Interrupt NOT Supported
9	1h RW	**Sys Addr 64bit Support (Sys_Addr_64bit_Support):** 1 - Core supports 64-bit System Address Bus. 0 - Core supports only 32-bit System Address Bus
8	1h RW	**Voltage Support 1_8V (Voltage_Support_1_8V):** 1: 1.8V Supported. 0: 1.8V NOT Supported
7	0h RW	**Voltage Support 3V (Voltage_Support_3V):** 1: 3.0V Supported. 0: 3.0V NOT Supported
6	0h RW	**Voltage Support 3_3V (Voltage_Support_3_3V):** 1: 3.3V Supported. 0: 3.3V NOT Supported
5	1h RW	**Suspend Resume Support (Suspend_Resume_Support):** 1: Suspend/Resume Supported. 0: Suspend/Resume NOT Supported
4	1h RW	**SDMA Support (SDMA_Support):** 1: SDMA mode Supported. 0: SDMA mode NOT Supported
3	1h RW	**High Speed Support (High_Speed_Support):** 1: HIGH_SPEED mode Supported. 0: HIGH_SPEED mode NOT Supported
2	1h RW	**ADMA2 Support (ADMA2_Support):** 1: ADMA2 mode Supported. 0: ADMA2 mode NOT Supported
1:0	0h RW	**Max Burst Length (Max_Burst_Length):** Maximum Block Length supported by the Core/Device. 00: 512 (Bytes). 01: 1024. 10: 2048. 11: Reserved

13.45 emmc Capabilities Bypass Register 2 (Cap_byps_reg2)—Offset 818h

eMMC Capabilities Bypass register 2

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 40040C8h

Bit Range	Default & Access	Field Name (ID): Description
31:27	0h RO	Reserved.
26:21	20h RW	**tuning count val (tuning_count_val):** Tuning Count Value Configures the Number of Taps (Phases) of the rxclk_in that is supported. The Tuning State machine uses this information to select one of the Taps (Phases) of the rxclk_in during the Tuning Procedure.
20	0h RW	**tuning dis (tuning_dis):** Disable the 1.5x Tuning count when calculating total tuning count. The internal tuning count will be set to the corecfg_Tuningcount when this signal is asserted
19	0h RW	**driver type 4 (driver_type_4):** Driver Type 4 Support 1b1 Supported 1b0 NOT Supported
18	0h RW	**driver type D (driver_type_D):** Driver Type D Support 1b1 Supported 1b0 NOT Supported
17	0h RW	**driver type C (driver_type_C):** Driver Type C Support 1b1 Supported 1b0 NOT Supported
16	0h RW	**driver type A (driver_type_A):** Driver Type A Support 1b1 Supported 1b0 NOT Supported
15	0h RO	Reserved.
14	1h RW	**support 8-bit embedded (support_8_bit_embedded):** 8-bit Support for Embedded Device 1b1 Supported 1b0 NOT Supported
13:8	0h RO	Reserved.
7:0	C8h RW	**base sd clock (base_sd_clock):** Base Clock Frequency for SD Clock

13.46 D0i3 (reg_D0i3)—Offset 81Ch

D0i3 Control register

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 8h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	Reserved.
3	1h RW/1C	**Restore Required (RestoreRequired):** When set (by HW), SW must restore state to the IP. The state may have been lost due to a reset or full power lost. SW clears the bit by writing a 1. This bit will be set on initial power up.
2	0h RW	**D0i3 (D0i3):** SW sets this bit to 1 to move the IP into the D0i3 state. Writing this bit to 0 will return the IP to the fully active D0 state (D0i0).
1	0h RO	**Interrupt Request (Interrupt_Request):** SW sets this bit to 1 to ask for an interrupt to be generated on completion of the command. SW must clear or set this on each write to this register. Not supported in SCS!
0	0h RO	**Cmd In Progress (Cmd_In_Progress):** HW sets this bit on a 1->0 or 0->1 transition of bit [2]. While set, the other bits in this register are not valid and it is illegal for SW to write to any bit in this register. When clear all the other bits in the register are valid and SW may write to any bit. If Interrupt Request bit [1] was set for the current command, HW may clear this bit before the interrupt has been made visible to SW, since when SW actually handles

13.47 Tx CMD Delay Control (Tx_CMD_dly)—Offset 820h

Front end module Tx Command Path Delay register

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 500h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	Reserved.
14:8	5h RW	**ddr mode (ddr_mode):** Tx CMD Delay (DDR Mode). 0 39 Select number of active delay elements. Each = 125pSec. 40 127 Reserved
7	0h RO	Reserved.
6:0	0h RW	**sdr mode (sdr_mode):** Tx CMD Delay (SDR Mode). 0 39 Select number of active delay elements. Each = 125pSec. 40 127 - Reserved

13.48 Tx DATA Delay Control 1 (Tx_DATA_dly_1)—Offset 824h

Front end module Tx data Path Delay register 1

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: A18h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	Reserved.
14:8	Ah RW	**hs400 mode (hs400_mode):** Tx Data Delay (HS400 Mode). 0 78 Select number of active delay elements. Each = 125pSec. 79 127 - Reserved
7	0h RO	Reserved.
6:0	18h RW	**sdr104 hs200 (sdr104_hs200):** Tx Data Delay (SDR104/HS200 Mode) 0-79 - Select the required delay, as a multiple of 125pSec.80 127 Reserved

13.49 Tx DATA Delay Control 2 (Tx_DATA_dly_2)—Offset 828h

Front end module Tx data Path Delay register 2

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 1C2A1C00h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	Reserved.
30:24	1Ch RW	**sdr50 mode (sdr50_mode):** Tx Data Delay (SDR50 Mode) 0-79 - Select the required delay, as a multiple of 125pSec. 80 127 Reserved
23	0h RO	Reserved.
22:16	2Ah RW	**ddr50 mode (ddr50_mode):** Tx Data Delay (DDR50 Mode). 0 78 Select number of active delay elements. Each = 125pSec. 79 127 - Reserved
15	0h RO	Reserved.

Bit Range	Default & Access	Field Name (ID): Description
14:8	1Ch RW	**sdr25 hs50 mode (sdr25_hs50_mode):** Tx Data Delay (SDR25/ HS50 Mode)0-79 - Select the required delay, as a multiple of 125pSec.80 127 Reserved
7	0h RO	Reserved.
6:0	0h RW	**sdr12 comp mode (sdr12_comp_mode):** Tx Data Delay (SDR12/Compatibility Mode) 0-79 - Select the required delay, as a multiple of 125pSec.80 127 Reserved

13.50 Rx CMD Data Delay Control 1 (Rx_CMD_Data_dly_1)—Offset 82Ch

Front end module Rx data Path Delay register 1

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: E0000h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	Reserved.
30:24	0h RW	**sdr50 mode (sdr50_mode):** Rx CMD + Data Delay (SDR50 Mode). 0 79 Select the required delay, as a multiple of 125pSec.80 127 Reserved
23	0h RO	Reserved.
22:16	Eh RW	**ddr50 mode (ddr50_mode):** Rx CMD + Data Delay (DDR50 Mode). 0 78 Select number of active delay elements. Each = 125pSec. 79 127 - Reserved
15	0h RO	Reserved.
14:8	0h RW	**sdr25 hs50 (sdr25_hs50):** Rx CMD + Data Delay (SDR25/HS50 Mode) 0-79 - Select the required delay, as a multiple of 125pSec.80 127 Reserved
7	0h RO	Reserved.
6:0	0h RW	**sdr12 comp (sdr12_comp):** Rx CMD + Data Delay (SDR12/ Compatibility Mode) 0-79 - Select the required delay, as a multiple of 125pSec.80 127 Reserved

13.51 Rx CMD Data Delay Control 2 (Rx_CMD_Data_dly_2)—Offset 834h

Front end module Rx data Path Delay register 2

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 10000h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	Reserved.
17:16	1h RW	**clk source (clk_source):** Clock Source for Rx Path 00 Rx Clock after Output Buffer 01 Rx Clock before Output Buffer 10 Automatic Selection based on Working mode (HS 200 before buffer, all others after buffer) 11 - Reserved
15:14	0h RO	Reserved.
13:8	0h RW	**path pll (path_pll):** Rx Path PLL #3 Delay value For Auto Tuning Mode 0-39 - Select the required delay, as a multiple of 125pSec.40 63 Reserved
7	0h RO	Reserved.
6:0	0h RW	**cmd data sdr104 hs200 (cmd_data_sdr104_hs200):** Rx CMD + Data Delay (SDR104/HS200 Mode) 0-79 - Select the required delay, as a multiple of 125pSec. 80 127 Reserved

13.52 Auto_tuning_val (Auto_tuning)—Offset 840h

Auto Tuning Value

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RO	Reserved.
4:0	0h RO/V	**Auto_tuning_val (Auto_tuning_val):** Auto tuning value

13.53 SDMA System Address Register/Argument2 Register (sdmasysaddr)—Offset 0h

This register contains concatinates reg_sdmasysaddrlo and reg_sdmasysaddrhi

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**sdma_sysaddress (sdma_sysaddress):** refer to reg_sdmasysaddrlo and reg_sdmasysaddrhi

13.54 BlockSize Register (blocksize)—Offset 4h

This register is used to configure the number of bytes in a data block

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	Reserved.
14:12	0h RW	**sdma_bufboundary (sdma_bufboundary):** This Field specifies DMA Buffer Boundary
11:0	0h RW	**xfer_blocksize (xfer_blocksize):** This Field specifies Block Size for Data Transfers

13.55 BlockCount Register (blockcount)—Offset 6h

This register is used to configure the number of data blocks

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW	**xfer_blockcount (xfer_blockcount):** This Register specifies Block Size for Data Transfers

13.56 Argument1 Register (argument1)—Offset 8h

This register contains concatinates argument1lo and argument1hi registers to result SD Command Argument

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**command_argument1 (command_argument1):** refer t oreg_argument1lo and reg_argument1hi.

13.57 TransferMode Register (transfermode)—Offset Ch

This register is used to control the operations of data transfers

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:6	0h RO	Reserved.
5	0h RW	**xfermode_multiblksel (xfermode_multiblksel):** This bit enables multiple block data transfers.
4	0h RW	**xfermode_dataxferdir (xfermode_dataxferdir):** This bit defines the direction of data transfers.
3:2	0h RW	**xfermode_autocmdena (xfermode_autocmdena):** This field determines use of auto command functions.
1	0h RW	**xfermode_blkcntena (xfermode_blkcntena):** This bit is used to enable the Block count register, which is only relevant for multiple block transfers
0	0h RW	**xfermode_dmaenable (xfermode_dmaenable):** '1' to enable DMA,

13.58 Command Register (command)—Offset Eh

This register is used to program the Command for host controller

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RO	Reserved.
13:8	0h RW	**command_cmdindex (command_cmdindex):** This bit shall be set to the command number (CMD0-63, ACMD0-63).
7:6	0h RW	**command_cmdtype (command_cmdtype):** This bit is used for select different command types.
5	0h RW	**command_datapresent (command_datapresent):** This bit is used to checking whether data present or not.
4	0h RW	**command_indexchkena (command_indexchkena):** This bit is used to enable/disable Command Index checking.
3	0h RW	**command_crcchkena (command_crcchkena):** This bit is used to enable/disable CRC checking.
2	0h RO	Reserved.
1:0	0h RW	**command_responsetype (command_responsetype):** Response Type Select.

13.59 Response Register (response01)—Offset 10h

This register is used to store responses from SD Cards (concatinates response0 & response1 registers)

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**command_response (command_response):** R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit

13.60 Response Register (response2)—Offset 14h

This register is used to store responses from SD Cards

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RO	**command_response (command_response):** R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit

13.61 Response Register (response3)—Offset 16h

This register is used to store responses from SD Cards

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RO	**command_response (command_response):** R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit

13.62 Response Register (response4)—Offset 18h

This register is used to store responses from SD Cards

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RO	**command_response (command_response):** R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit

13.63 Response Register (response5)—Offset 1Ah

This register is used to store responses from SD Cards

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RO	**command_response (command_response):** R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit

13.64 Response Register (response6)—Offset 1Ch

This register is used to store responses from SD Cards

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RO	**command_response (command_response):** R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit

13.65 Response Register (response7)—Offset 1Eh

This register is used to store responses from SD Cards

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RO	**command_response (command_response):** R[] refers to a bit range within the response data as transmitted on the SD Bus, REP[] refers to a bit

13.66 Buffer DataPort Register (dataport)—Offset 20h

This register is used to access internal buffer

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**sdhcdmactrl_piobufrddata (sdhcdmactrl_piobufrddata):** The Host Controller Buffer can be accessed through this 32-bit Data Port Register.

13.67 PresentState Register (presentstate)—Offset 24h

The Host Driver can get status of the Host Controller from this 32-bit read-only register

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	Reserved.
28	0h RO	**sdif_dat7in_dsync (sdif_dat7in_dsync):** This status is used to check DAT line level to recover from errors, and for debugging.
27	0h RO	**sdif_dat6in_dsync (sdif_dat6in_dsync):** This status is used to check DAT line level to recover from errors, and for debugging.
26	0h RO	**sdif_dat5in_dsync (sdif_dat5in_dsync):** This status is used to check DAT line level to recover from errors, and for debugging.
25	0h RO	**sdif_dat4in_dsync (sdif_dat4in_dsync):** This status is used to check DAT line level to recover from errors, and for debugging.
24	0h RO	**sdif_cmdin_dsync (sdif_cmdin_dsync):** This status is used to check DAT line level to recover from errors, and for debugging.
23	0h RO	**sdif_dat3in_dsync (sdif_dat3in_dsync):** This status is used to check DAT line level to recover from errors, and for debugging. This is
22	0h RO	**sdif_dat2in_dsync (sdif_dat2in_dsync):** This status is used to check DAT line level to recover from errors, and for debugging. This is

Bit Range	Default & Access	Field Name (ID): Description
21	0h RO	**sdif_dat1in_dsync (sdif_dat1in_dsync):** This status is used to check DAT line level to recover from errors, and for debugging. This is
20	0h RO	**sdif_dat0in_dsync (sdif_dat0in_dsync):** This status is used to check DAT line level to recover from errors, and for debugging. This is
19	0h RO	**sdif_wp_dsync (sdif_wp_dsync):** The Write Protect Switch is supported for memory and combo cards.
18	0h RO	**sdif_cd_n_dsync (sdif_cd_n_dsync):** This bit reflects the inverse value of the SDCD# pin.
17	0h RO	**sdhccarddet_statestable_dsync (sdhccarddet_statestable_dsync):** This bit is used for testing.
16	0h RO	**sdhccarddet_inserted_dsync (sdhccarddet_inserted_dsync):** This bit indicates whether a card has been inserted.
15:12	0h RO	Reserved.
11	0h RO	**sdhcdmactrl_piobufrdena (sdhcdmactrl_piobufrdena):** This status is used for non-DMA read transfers. This read only flag indicates that valid data exists in the host side buffer status.
10	0h RO	**sdhcdmactrl_piobufwrena (sdhcdmactrl_piobufwrena):** This status is used for non-DMA write transfers.This read only flag indicates if space is available for write data.
9	0h RO	**sdhcdmactrl_rdxferactive (sdhcdmactrl_rdxferactive):** This status is used for detecting completion of a read transfer.
8	0h RO	**sdhcdmactrl_wrxferactive (sdhcdmactrl_wrxferactive):** This status indicates a write transfer is active.
7:4	0h RO	Reserved.
3	0h RO	**sdhcsdctrl_retuningreq_dsync (sdhcsdctrl_retuningreq_dsync):** Host Controller may request Host Driver to execute re-tuning sequence by setting this bit.
2	0h RO	**sdhcdmactrl_datalineactive (sdhcdmactrl_datalineactive):** This bit indicates whether one of the DAT line on SD bus is in use.
1	0h RO	**presentstate_inhibitdat (presentstate_inhibitdat):** This status bit is generated if either the DAT Line Active or the Read transfer Active is set to 1.
0	0h RO	**presentstate_inhibitcmd (presentstate_inhibitcmd):** If this bit is 0, it indicates the CMD line is not in use and the HC can issue a SD command using the CMD line. This bit is set immediately after the Command register (00Fh) is written. This bit is cleared when the command response is received.

13.68 HostControl1 Register (hostcontrol1)—Offset 28h

This register is used to program DMA modes, LED Control, Data Transfer Width, High Speed Enable, Card detect test level and signal selection

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**hostctrl1_cdsigselect (hostctrl1_cdsigselect):** This bit selects source for card detection.
6	0h RW	**hostctrl1_cdtestlevel (hostctrl1_cdtestlevel):** This bit is used for indicating whether card is inserted or not.
5	0h RW	**hostctrl1_extdatawidth (hostctrl1_extdatawidth):** This bit controls 8-bit bus width mode for embedded device.
4:3	0h RW	**hostctrl1_dmaselect (hostctrl1_dmaselect):** One of supported DMA modes can be selected.
2	0h RW	**hostctrl1_highspeedena (hostctrl1_highspeedena):** This bit is used for setting speed mode.
1	0h RW	**hostctrl1_datawidth (hostctrl1_datawidth):** This bit selects the data width of the HC.
0	0h RW	**hostctrl1_ledcontrol (hostctrl1_ledcontrol):** This bit is used to caution the user not to remove the card while the SD card is being accessed.

13.69 PowerControl Register (powercontrol)—Offset 29h

This register is used to program the SD Bus power and voltage level

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:5	0h RO	Reserved.
4	0h RW	**emmc_hwreset (emmc_hwreset):** Hardware reset signal is generated for eMMC card.
3:1	0h RW	**pwrctrl_sdbusvoltage (pwrctrl_sdbusvoltage):** By setting these bits, the HD selects the voltage level for the SD card.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**pwrctrl_sdbuspower (pwrctrl_sdbuspower):** This bit is used detect whether power is on or off.

13.70 BlockGapControl Register (blockgapcontrol)—Offset 2Ah

This register is used to program the block gap request, read wait control and interrupt at block gap

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 80h

Bit Range	Default & Access	Field Name (ID): Description
7	1h RW	**blkgapctrl_bootackena (blkgapctrl_bootackena):** To check for the boot acknowledge in boot operation.
6	0h RW	**blkgapctrl_altbootmode (blkgapctrl_altbootmode):** To start boot code access in alternative mode.
5	0h RW	**blkgapctrl_bootenable (blkgapctrl_bootenable):** To start boot code access. '0' To stop boot code access, '1' To start boot code access
4	0h RW	**blkgapctrl_spimode (blkgapctrl_spimode):** SPI mode enable bit. '0' SD Mode, '1' SPI Mode
3	0h RW	**blkgapctrl_interrupt (blkgapctrl_interrupt):** This bit is valid only in 4-bit mode of the SDIO card andselects a sample point in the interrupt cycle.
2	0h RW	**blkgapctrl_rdwaitctrl (blkgapctrl_rdwaitctrl):** The read wait function is optional for SDIO cards.
1	0h RW	**blkgapctrl_continue (blkgapctrl_continue):** This bit is used to restart a transaction which was stopped using the Stop At Block Gap Request.
0	0h RW	**blkgapctrl_stopatblkgap (blkgapctrl_stopatblkgap):** This bit is used to stop executing a transaction at the next block gap for non- DMA,SDMA and ADMA transfers.

13.71 Wakeup Control Register (wakeupcontrol)—Offset 2Bh

This register is used to program the wakeup functionality

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:3	0h RO	Reserved.
2	0h RW	**wkupctrl_cardremoval (wkupctrl_cardremoval):** This bit enables wakeup event via Card Insertion assertion in the Normal Interrupt Status register.
1	0h RW	**wkupctrl_cardinsertion (wkupctrl_cardinsertion):** This bit enables wakeup event via Card Insertion assertion in the Normal Interrupt Status register.
0	0h RW	**wkupctrl_cardinterrupt (wkupctrl_cardinterrupt):** This bit enables wakeup event via Card Interrupt assertion in the Normal Interrupt Status register.

13.72 Clock Control Register (clockcontrol)—Offset 2Ch

This register is used to program the Clock frequency select, generator select, Clock enable, Internal Clock state fields

Access Method

Type: MEM Register (Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RW	**clkctrl_sdclkfreqsel (clkctrl_sdclkfreqsel):** This register is used to select the frequency of the SDCLK pin.
7:6	0h RW	**clkctrl_sdclkfreqsel_upperbits (clkctrl_sdclkfreqsel_upperbits):** Bit 07-06 is assigned to bit 09-08 of clock divider in SDCLK Frequency Select
5	0h RW	**clkctrl_clkgensel (clkctrl_clkgensel):** This bit is used to select the clock generator mode in SDCLK Frequency Select.
4:3	0h RO	Reserved.
2	0h RW	**clkctrl_sdclkena (clkctrl_sdclkena):** This bit enables/disables SD Clock. '0' Disable, '1' Enable
1	0h RW	**sdhcclkgen_intclkstable_dsync (sdhcclkgen_intclkstable_dsync):** This bit is set to 1 when SD clock is stable after writing to Internal Clock Enable in this register to 1.
0	0h RW	**clkctrl_intclkena (clkctrl_intclkena):** This bit enables setting of internal clock.'0' Stop, '1' Oscillate

13.73 Timeout Control Register (timeoutcontrol)—Offset 2Eh

The register sets the Data Timeout counter value

Access Method

Type: MEM Register (Size: 8 bits)	**Device:** **Function:**

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:4	0h RO	Reserved.
3:0	0h RW	**timeout_ctrvalue (timeout_ctrvalue):** This value determines the interval by which DAT line time-outs are detected.

13.74 Software Reset Register (softwarereset)—Offset 2Fh

This register is used to program the software reset for data, command and for all.

Access Method

Type: MEM Register (Size: 8 bits)	**Device:** **Function:**

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:3	0h RO	Reserved.
2	0h RW	**swreset_for_dat (swreset_for_dat):** Only part of data circuit is reset. '0' Work, '1' Reset
1	0h RW	**swreset_for_cmd (swreset_for_cmd):** Only part of command circuit is reset.'0' Work, '1' reset.
0	0h RW	**swreset_for_all (swreset_for_all):** If this bit is set to 1, the SD card shall reset itself and must be re initialized by the HD.

13.75 Normal Interrupt Status Register (normalintrsts)—Offset 30h

This register gives the status of all the interrupts

Access Method

Type: MEM Register (Size: 16 bits)	**Device:** **Function:**

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**reg_errorintrsts (reg_errorintrsts):** This bit is set If any of the bits in the Error Interrupt Status Register are set.
14	0h RW/1C	**normalintrsts_bootcomplete (normalintrsts_bootcomplete):** This status is set if the boot operation gets terminated.
13	0h RW/1C	**normalintrsts_rcvbootack (normalintrsts_rcvbootack):** This status is set if the boot acknowledge is received from device.
12	0h RO	**normalintrsts_retuningevent (normalintrsts_retuningevent):** This status is set if Re-Tuning Request in the Present State register changes from 0 to 1.
11	0h RO	**normalintrsts_intc (normalintrsts_intc):** This status is set if INT_C is enabled and INT_C# pin is in low level. It is cleared by resetting the INT_C interrupt factor.
10	0h RO	**normalintrsts_intb (normalintrsts_intb):** This status is set if INT_B is enabled and INT_B# pin is in low level. It is cleared by resetting
9	0h RW	**normalintrsts_inta (normalintrsts_inta):** This status is set if INT_A is enabled and INT_A# pin is in low level. It is cleared by resetting
8	0h RW	**normalintrsts_cardintsts (normalintrsts_cardintsts):** This status is set to generate Card Interrupt. '0' No Card Interrupt, '1' Generate Card Interrupt
7	0h RW/1C	**normalintrsts_cardremsts (normalintrsts_cardremsts):** This status is set if the Card Inserted in the Present State register changes from 1 to 0.
6	0h RW/1C	**normalintrsts_cardinssts (normalintrsts_cardinssts):** This status is set if the Card Inserted in the Present State register changes from 0 to 1.
5	0h RW/1C	**normalintrsts_bufrdready (normalintrsts_bufrdready):** This status is set if the Buffer Read Enable changes from 0 to 1.
4	0h RW/1C	**normalintrsts_bufwrready (normalintrsts_bufwrready):** This status is set if the Buffer Write Enable changes from 0 to 1.
3	0h RW/1C	**normalintrsts_dmainterrupt (normalintrsts_dmainterrupt):** This status is set if the HC detects the Host DMA Buffer Boundary in the Block Size regiser.
2	0h RW/1C	**normalintrsts_blkgapevent (normalintrsts_blkgapevent):** If the Stop At Block Gap Request in the BlockGap Control Register is set, this bit is set.
1	0h RW/1C	**normalintrsts_xfercomplete (normalintrsts_xfercomplete):** This bit is set when a read / write transaction is completed.
0	0h RW/1C	**normalintrsts_cmdcomplete (normalintrsts_cmdcomplete):** This bit is set when we get the end bit of the command response.

13.76 ErrorInterruptStatus_Register (errorintrsts)—Offset 32h

This register gives the status of the error interrupts

Access Method

Type: MEM Register (Size: 16 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	Reserved.
12	0h RW/1C	**errorintrsts_hosterror (errorintrsts_hosterror):** Occurs when detecting Host ERROR
11:10	0h RO	Reserved.
9	0h RW/1C	**errorintrsts_admaerror (errorintrsts_admaerror):** This bit is set when the Host Controller detects errors during ADMA based data transfer.
8	0h RW/1C	**errorintrsts_autocmderror (errorintrsts_autocmderror):** This bit is set when detecting that one of the bits D00-D04 in Auto CMD Error Status register has changed from 0 to 1.
7	0h RW/1C	**errorintrsts_currlimiterror (errorintrsts_currlimiterror):** If this bit is 1, it means that the HC is not supplying power to SD card due to some failure.Reading 0 means that the HC is supplying power and no error has occurred.
6	0h RW/1C	**errorintrsts_dataendbiterror (errorintrsts_dataendbiterror):** Occurs when detecting 0 at the end bit position of read data which uses the DAT line or the end bit position of the CRC status.
5	0h RW/1C	**errorintrsts_datacrcerror (errorintrsts_datacrcerror):** Occurs when detecting CRC error when transferring read data which uses the DAT line or when detecting the Write CRC Status having a value of other than 010.
4	0h RW/1C	**errorintrsts_datatimeouterror (errorintrsts_datatimeouterror):** This status is set if Data Timeout error occurs.
3	0h RW/1C	**errorintrsts_cmdindexerror (errorintrsts_cmdindexerror):** Occurs if a Command Index error occurs in the Command Response.
2	0h RW/1C	**errorintrsts_cmdendbiterror (errorintrsts_cmdendbiterror):** Occurs when detecting that the end bit of a command response is 0.
1	0h RW/1C	**errorintrsts_cmdcrcerror (errorintrsts_cmdcrcerror):** This bit is set when Command CRC Error is generated. '0' No Error, '1' CRC Error

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW/1C	**errorintrsts_cmdtimeouterror (errorintrsts_cmdtimeouterror):** This bit is set if no response is returned within 64 SDCLK cycles from the end bit of the command.

13.77 Normal Interrupt Status Enable Register (normalintrstsena)—Offset 34h

This register is used to enable the normal interrupt status register fields

Access Method

Type: MEM Register (Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**normalintrsts_enableregbit15 (normalintrsts_enableregbit15):** The HC shall control error Interrupts using the Error Interrupt Status Enable register.
14	0h RW	**normalintrsts_enableregbit14 (normalintrsts_enableregbit14):** This status is set if the boot operation gets terminated.
13	0h RW	**normalintrsts_enableregbit13 (normalintrsts_enableregbit13):** This status is set if the boot acknowledge is received from device.
12	0h RW	**normalintrsts_enableregbit12 (normalintrsts_enableregbit12):** This status is set if Re-Tuning Request in the Present State register changes from 0 to 1.
11	0h RW	**normalintrsts_enableregbit11 (normalintrsts_enableregbit11):** If this bit is set to 0, the Host Controller shall clear the interrupt request to the System.
10	0h RW	**normalintrsts_enableregbit10 (normalintrsts_enableregbit10):** If this bit is set to 0, the Host Controller shall clear the interrupt request to the System.
9	0h RW	**normalintrsts_enableregbit9 (normalintrsts_enableregbit9):** If this bit is set to 0, the Host Controller shall clear the interrupt request to the System.
8	0h RW	**sdhcregset_cardintstsena (sdhcregset_cardintstsena):** If this bit is set to 0, the HC shall clear Interrupt request to the System. The Card Interrupt detection is stopped when this bit is cleared and restarted when this bit is set to 1.
7	0h RW	**sdhcregset_cardremstsena (sdhcregset_cardremstsena):** This status is set if the Card Inserted in the Present State register changes from 1 to 0.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**sdhcregset_cardinsstsena (sdhcregset_cardinsstsena):** This status is set if the Card Inserted in the Present State register changes from 0 to 1.
5	0h RW	**normalintrsts_enableregbit5 (normalintrsts_enableregbit5):** This status is set if the Buffer Read Enable changes from 0 to 1.
4	0h RW	**normalintrsts_enableregbit4 (normalintrsts_enableregbit4):** This status is set if the Buffer Write Enable changes from 0 to 1.
3	0h RW	**normalintrsts_enableregbit3 (normalintrsts_enableregbit3):** This status is set if the HC detects the Host DMA Buffer Boundary in the Block Size regiser.
2	0h RW	**normalintrsts_enableregbit2 (normalintrsts_enableregbit2):** If the Stop At Block Gap Request in the BlockGap Control Register is set, this bit is set.
1	0h RW	**normalintrsts_enableregbit1 (normalintrsts_enableregbit1):** This bit is set when a read / write transaction is completed.
0	0h RW	**normalintrsts_enableregbit0 (normalintrsts_enableregbit0):** This bit is set when we get the end bit of the command response.

13.78 Error Interrupt Status Enable Register (errorintrstsena)—Offset 36h

This register is used to enable the Error Interrupt Status register fields

Access Method

Type: MEM Register (Size: 16 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	Reserved.
12	0h RW	**errorintrsts_enableregbit12 (errorintrsts_enableregbit12):** Occurs when detecting ERROR in m_hresp(dma transaction)
11	0h RO	Reserved.
10	0h RW	**errorintrsts_enableregbit10 (errorintrsts_enableregbit10):** This status is set if INT_B is enabled and INT_B# pin is in low level. It is cleared by resetting

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**errorintrsts_enableregbit9 (errorintrsts_enableregbit9):** This bit is set when the Host Controller detects errors during ADMA based data transfer.
8	0h RW	**errorintrsts_enableregbit8 (errorintrsts_enableregbit8):** This bit is set when detecting that one of the bits D00-D04 in Auto CMD Error Status register has changed from 0 to 1.
7	0h RW	**errorintrsts_enableregbit7 (errorintrsts_enableregbit7):** If this bit is 1, it means that the HC is not supplying power to SD card due to some failure.Reading 0 means that the HC is supplying power and no error has occurred.
6	0h RW	**errorintrsts_enableregbit6 (errorintrsts_enableregbit6):** Occurs when detecting 0 at the end bit position of read data which uses the DAT line or the end bit position of the CRC status.
5	0h RW	**errorintrsts_enableregbit5 (errorintrsts_enableregbit5):** Occurs when detecting CRC error when transferring read data which uses the DAT line or when detecting the Write CRC Status having a value of other than 010.
4	0h RW	**errorintrsts_enableregbit4 (errorintrsts_enableregbit4):** This status is set if Data Timeout error occurs.
3	0h RW	**errorintrsts_enableregbit3 (errorintrsts_enableregbit3):** Occurs if a Command Index error occurs in the Command Response.
2	0h RW	**errorintrsts_enableregbit2 (errorintrsts_enableregbit2):** Occurs when detecting that the end bit of a command response is 0.
1	0h RW	**errorintrsts_enableregbit1 (errorintrsts_enableregbit1):** This bit is set when Command CRC Error is generated.
0	0h RW	**errorintrsts_enableregbit0 (errorintrsts_enableregbit0):** This bit is set if no response is returned within 64 SDCLK cycles from the end bit of the command.

13.79 Normal Interrupt Signal Enable Register (normalintrsigena)—Offset 38h

This register is used to enable the Normal Interrupt Signal register

Access Method

Type: MEM Register (Size: 16 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**normalintrsts_enableregbit15 (normalintrsts_enableregbit15):** The HD shall control error Interrupts using the Error Interrupt Signal Enable register.
14	0h RW	**normalintrsts_enableregbit14 (normalintrsts_enableregbit14):** Boot Terminate Interrupt Signal Enable
13	0h RW	**normalintrsts_enableregbit13 (normalintrsts_enableregbit13):** Boot ack rcv Signal Enable
12	0h RW	**normalintrsts_enableregbit12 (normalintrsts_enableregbit12):** Re-Tuning Event Signal Enable
11	0h RW	**normalintrsts_enableregbit11 (normalintrsts_enableregbit11):** INT_C Signal Enable
10	0h RW	**normalintrsts_enableregbit10 (normalintrsts_enableregbit10):** INT_B Signal Enable
9	0h RW	**normalintrsts_enableregbit9 (normalintrsts_enableregbit9):** INT_A Signal Enable
8	0h RW	**sdhcregset_cardintstsena (sdhcregset_cardintstsena):** Card Interrupt Signal Enable
7	0h RW	**sdhcregset_cardremstsena (sdhcregset_cardremstsena):** Card Removal Signal Enable
6	0h RW	**sdhcregset_cardinsstsena (sdhcregset_cardinsstsena):** Card Insertion Signal Enable
5	0h RW	**normalintrsts_enableregbit5 (normalintrsts_enableregbit5):** Buffer Read Ready Signal Enable
4	0h RW	**normalintrsts_enableregbit4 (normalintrsts_enableregbit4):** Buffer Write Ready Signal Enable
3	0h RW	**normalintrsts_enableregbit3 (normalintrsts_enableregbit3):** DMA Interrupt Signal Enable
2	0h RW	**normalintrsts_enableregbit2 (normalintrsts_enableregbit2):** Block Gap Event Signal Enable
1	0h RW	**normalintrsts_enableregbit1 (normalintrsts_enableregbit1):** Transfer Complete Signal Enable
0	0h RW	**normalintrsts_enableregbit0 (normalintrsts_enableregbit0):** Command Complete Signal Enable

13.80 Error Interrupt Signal Enable Register (errorintrsigena)—Offset 3Ah

This register is used to enable Error Interrupt Signal register

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	Reserved.
12	0h RO	**errorintrsig_enableregbit12 (errorintrsig_enableregbit12):** Target Response Error Signal Enable
11	0h RO	Reserved.
10	0h RW	**errorintrsig_enableregbit10 (errorintrsig_enableregbit10):** Tuning Error Signal Enable
9	0h RW	**errorintrsig_enableregbit9 (errorintrsig_enableregbit9):** ADMA Error Signal Enable
8	0h RW	**errorintrsig_enableregbit8 (errorintrsig_enableregbit8):** Auto CMD Error Signal Enable
7	0h RW	**errorintrsig_enableregbit7 (errorintrsig_enableregbit7):** Current Limit Error Signal Enable
6	0h RW	**errorintrsig_enableregbit6 (errorintrsig_enableregbit6):** Data End Bit Error Signal Enable
5	0h RW	**errorintrsig_enableregbit5 (errorintrsig_enableregbit5):** Data CRC Error Signal Enable
4	0h RW	**errorintrsig_enableregbit4 (errorintrsig_enableregbit4):** Data Timeout Error Signal Enable
3	0h RW	**errorintrsig_enableregbit3 (errorintrsig_enableregbit3):** Command Index Error Signal Enable
2	0h RW	**errorintrsig_enableregbit2 (errorintrsig_enableregbit2):** Command End Bit Error Signal Enable
1	0h RW	**errorintrsig_enableregbit1 (errorintrsig_enableregbit1):** Command CRC Error Signal Enable
0	0h RW	**errorintrsig_enableregbit0 (errorintrsig_enableregbit0):** Command Timeout Error Signal Enable

13.81 Auto CMD12 Error Status Register (autocmderrsts)—Offset 3Ch

This register is used to indicate CMD12 response error of Auto CMD12 and CMD23 response error of Auto CMD 23

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	Reserved.
7	0h RO	**autocmderrsts_nexterror (autocmderrsts_nexterror):** Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 error(D04- D01) in this register. This bit is set to 0 when Auto CMD Error is generated by Auto CMD23
6:5	0h RO	Reserved.
4	0h RO	**autocmderrsts_indexerror (autocmderrsts_indexerror):** Occurs if the Command Index error occurs in response to a command.
3	0h RO	**autocmderrsts_endbiterror (autocmderrsts_endbiterror):** Occurs when detecting that the end bit of command response is 0.
2	0h RO	**autocmderrsts_crcerror (autocmderrsts_crcerror):** Occurs when detecting a CRC error in the command response.
1	0h RO	**autocmderrsts_timeouterror (autocmderrsts_timeouterror):** Occurs if the no response is returned within 64 SDCLK cycles from the end bit of the command.
0	0h RO	**autocmderrsts_notexecerror (autocmderrsts_notexecerror):** Setting this bit to 1 means the HC cannot issue Auto CMD12 to stop memory multiple block transfer due to some error.

13.82 Host Control2 Register (hostcontrol2)—Offset 3Eh

This register is used to program UHS Select Mode,UHS Select Mode,Driver Strength Select,Execute Tuning,Sampling Clock Select,Asynchronous Interrupt Enable and Preset value enable

Access Method

Type: MEM Register (Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**hostctrl2_presetvalueenable (hostctrl2_presetvalueenable):** This bit enables the functions defined in the Preset Value registers.
14	0h RW	**hostctrl2_asynchintrenable (hostctrl2_asynchintrenable):** This bit can be set to 1 if a card support asynchronous interrupt and Asynchronous Interrupt Support is set to 1 in the Capabilities register.

Bit Range	Default & Access	Field Name (ID): Description
13:10	0h RO	Reserved.
9	0h RW	**hostctrl2_driverstrength_bit2 (hostctrl2_driverstrength_bit2):** This is the programmed Drive STrength output and Bit[2] of the sdhccore_drivestrength value.
8	0h RO	Reserved.
7	0h RW	**hostctrl2_samplingclkselect (hostctrl2_samplingclkselect):** This bit is set by tuning procedure when Execute Tuning is cleared.
6	0h RW	**hostctrl2_executetuning (hostctrl2_executetuning):** This bit is set to 1 to start tuning procedure and automatically cleared when tuning procedure is completed.
5:4	0h RW	**hostctrl2_driverstrength (hostctrl2_driverstrength):** Host Controller output driver in 1.8V signaling is selected by this bit.
3	0h RW	**hostctrl2_1p8vsignallingena (hostctrl2_1p8vsignallingena):** This bit controls voltage regulator for I/O cell.
2:0	0h RW	**hostctrl2_uhsmodeselect (hostctrl2_uhsmodeselect):** This field is used to select one of UHS-I modes and effective when 1.8V Signaling Enable is set to 1.

13.83 Capabilities Register (capabilities)—Offset 40h

This register provides the host driver with information specific to the host controller implementation

Access Method

Type: MEM Register (Size: 64 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RO	**corecfg_hs400support (corecfg_hs400support):** This field indicates whether HS400 is supported or not.
62:58	0h RO	Reserved.
57	0h RO	**corecfg_spiblkmode (corecfg_spiblkmode):** This field indicates whether SPI Block Mode is supported or not.
56	0h RO	**corecfg_spisupport (corecfg_spisupport):** This field indicates whether SPI Mode is supported or not.
55:48	0h RO	**corecfg_clockmultiplier (corecfg_clockmultiplier):** This field indicates clock multiplier value of programmable clock generator.
47:46	0h RO	**corecfg_retuningmodes (corecfg_retuningmodes):** This field defines the re-tuning capability of a Host Controller.

Bit Range	Default & Access	Field Name (ID): Description
45	0h RO	**corecfg_tuningforsdr50 (corecfg_tuningforsdr50):** If this bit is set to 1, this Host Controller requires tuning to operate SDR50.
44	0h RO	Reserved.
43:40	0h RO	**corecfg_retuningtimercnt (corecfg_retuningtimercnt):** This field indicates an initial value of the Re-Tuning Timer for Re-Tuning Mode 1 to 3.
39	0h RO	**corecfg_type4support (corecfg_type4support):** This bit indicates support of Type 4 Driver.
38	0h RO	**corecfg_ddriversupport (corecfg_ddriversupport):** This bit indicates support of Driver Type D for 1.8 Signaling.
37	0h RO	**corecfg_cdriversupport (corecfg_cdriversupport):** This bit indicates support of Driver Type C for 1.8 Signaling.
36	0h RO	**corecfg_adriversupport (corecfg_adriversupport):** This bit indicates support of Driver Type A for 1.8 Signaling.
35	0h RO	Reserved.
34	0h RO	**corecfg_ddr50support (corecfg_ddr50support):** This bit indicates whether DDR50 is supported.
33	0h RO	**corecfg_sdr104support (corecfg_sdr104support):** This bit indicates whether SDR104 is supported.SDR104 requires tuning.
32	0h RO	**corecfg_sdr50support (corecfg_sdr50support):** This bit indicates whether SDR50 is supported.
31:30	0h RO	**corecfg_slottype (corecfg_slottype):** This field indicates usage of a slot by a specific Host System
29	0h RO	**corecfg_asynchintrsupport (corecfg_asynchintrsupport):** Asynchronous Interrupt Support
28	0h RO	**corecfg_64bitsupport (corecfg_64bitsupport):** This bit indicates whether the HC supports 64bit System Bus
27	0h RO	Reserved.
26	0h RO	**corecfg_1p8voltsupport (corecfg_1p8voltsupport):** This bit indicates whether the HC supports 1.8V.
25	0h RO	**corecfg_3p0voltsupport (corecfg_3p0voltsupport):** This bit indicates whether the HC supports 3.0V.
24	0h RO	**corecfg_3p3voltsupport (corecfg_3p3voltsupport):** This bit indicates whether the HC supports 3.3V.
23	0h RO	**corecfg_suspressupport (corecfg_suspressupport):** This bit indicates whether the HC supports Suspend/Resume functionality.
22	0h RO	**corecfg_sdmasupport (corecfg_sdmasupport):** This bit indicates whether the HC is capable of using DMA to transfer data between system memory and the HC directly.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RO	**corecfg_highspeedsupport (corecfg_highspeedsupport):** This bit indicates whether the HC and the Host System support High Speed mode.
20	0h RO	Reserved.
19	0h RO	**corecfg_adma2support (corecfg_adma2support):** '0'ADMA2 Not Supported
18	0h RO	**corecfg_8bitsupport (corecfg_8bitsupport):** This bit indicates whether the Host Controller is capable of using 8-bit bus width mode.
17:16	0h RO	**corecfg_maxblklength (corecfg_maxblklength):** This value indicates the maximum block size that the HD can read and write to the buffer in the HC.
15:8	0h RO	**corecfg_baseclkfreq (corecfg_baseclkfreq):** 6-bit Base Clock Frequency
7	0h RO	**corecfg_timeoutclkunit (corecfg_timeoutclkunit):** This bit shows the unit of base clock frequency used to detect Data Timeout Error.
6	0h RO	Reserved.
5:0	0h RO	**corecfg_timeoutclkfreq (corecfg_timeoutclkfreq):** This bit shows the base clock frequency used to detect Data Timeout Error.

13.84 Maximum Current Capabilities Register (maxcurrentcap)—Offset 48h

This register indicates maximum current capability for each voltage

Access Method

Type: MEM Register (Size: 64 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:24	0h RO	Reserved.
23:16	0h RO	**corecfg_maxcurrent1p8v (corecfg_maxcurrent1p8v):** Maximum Current for 1.8V
15:8	0h RO	**corecfg_maxcurrent3p0v (corecfg_maxcurrent3p0v):** Maximum Current for 3.0V
7:0	0h RO	**corecfg_maxcurrent3p3v (corecfg_maxcurrent3p3v):** Maximum Current for 3.3V

13.85 Force Event REGISTER for AUTO CMD Error Status (ForceEventforAUTOCMDErrorStatus)—Offset 50h

This register is not physically implemented, rather it is an address where Auto CMD Error Status register can be written.

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	Reserved.
7	0h RO	**forcecmdnotissuedbyautocmd12err (forcecmdnotissuedbyautocmd12err):** Force Event for Command Not Issued by AUTO CMD12 Error
6:5	0h RO	Reserved.
4	0h RO	**forceautocmdindexerr (forceautocmdindexerr):** Force Event for AUTO CMD Index Error
3	0h RO	**forceautocmdendbiterr (forceautocmdendbiterr):** Force Event for AUTO CMD End Bit Error
2	0h RO	**forceautocmdcrcerr (forceautocmdcrcerr):** Force Event for AUTO CMD Timeout Error
1	0h RO	**forceautocmdtimeouterr (forceautocmdtimeouterr):** Force Event for AUTO CMD Timeout Error
0	0h RO	**forceautocmdnotexec (forceautocmdnotexec):** Force Event for AUTO CMD12 Not Executed

13.86 Force Event Register for Error Interrupt Status (forceeventforerrintsts)—Offset 52h

This register is not physically implemented, rather it is an address where Error Interrupt Status register can be written.

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:11	0h RO	Reserved.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RO	**forcetuningerr (forcetuningerr):** Force Event for Tuning Error
9	0h RO	**forceadmaerr (forceadmaerr):** Force Event for ADMA Error'0' no interrupt, '1' interrupt generated
8	0h RO	**forceautocmderr (forceautocmderr):** Force Event for Auto CMD Error '0' no interrupt, '1' interrupt generated
7	0h RO	**forcecurrlimerr (forcecurrlimerr):** Force Event for Current Limit Error '0' no interrupt, '1' interrupt generated
6	0h RO	**forcedatendbiterr (forcedatendbiterr):** Force Event for Data End Bit Error. '0' no interrupt, '1' interrupt generated
5	0h RO	**forcedatcrcerr (forcedatcrcerr):** Force Event for Data CRC Error
4	0h RO	**forcedattimeouterr (forcedattimeouterr):** Force Event for Data Timeout Error '0' no interrupt, '1' interrupt generated
3	0h RO	**forcecmdindexerr (forcecmdindexerr):** Force Event for Command Index Error
2	0h RO	**forcecmdendbiterr (forcecmdendbiterr):** Force Event for Command End Bit Error '0' no interrupt, '1' interrupt generated
1	0h RO	**forcecmdcrcerr (forcecmdcrcerr):** Force Event for Command CRC Error
0	0h RO	**forcecmdtimeouterr (forcecmdtimeouterr):** Force Event for CMD Timeout Error '0' No interrupt, '1' interrupt generated

13.87 ADMA Error Status Register (admaerrsts)—Offset 54h

When the ADMA Error interrupt occur, this register holds the ADMA State in ADMA Error States field and ADMA System Address holds address around the error descriptor

Access Method

Type: MEM Register (Size: 8 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:3	0h RO	Reserved.
2	0h RO	**admaerrsts_admalenmismatcherr (admaerrsts_admalenmismatcherr):** ADMA Length Mismatch Error
1:0	0h RO	**admaerrsts_admaerrorstate (admaerrsts_admaerrorstate):** This field indicates the state of ADMA when error is occurred during ADMA data transfer.

13.88 ADMA System Address Register0&1 (admasysaddr01)—Offset 58h

This register contains the physical address used for ADMA data transfer

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**adma_sysaddress0 (adma_sysaddress0):** This register holds byte address of executing command of the Descriptor table.

13.89 ADMA System Address Register1 (admasysaddr2)—Offset 5Ch

This register contains the physical address used for ADMA data transfer

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW	**adma_sysaddress2 (adma_sysaddress2):** This register holds byte address of executing command of the Descriptor table.

13.90 ADMA System Address Register1 (admasysaddr3)—Offset 5Eh

This register contains the physical address used for ADMA data transfer

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW	**adma_sysaddress3 (adma_sysaddress3):** This register holds byte address of executing command of the Descriptor table.

13.91 Preset Value Register for Initialization (presetvalue0)—Offset 60h

This register is used to read the SDCLK Frequency Select Value,Clock Generator Select Value,Driver Strength Select Value

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 4h

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RO	**DriverStrengthSelectValue (DriverStrengthSelectValue):** Driver Strength Select Value
13:11	0h RO	Reserved.
10	0h RO	**ClockGeneratorSelectValue (ClockGeneratorSelectValue):** This bit is effective when Host Controller supports programmable clock
9:0	4h RO	**SDCLKFrequencySelectValue (SDCLKFrequencySelectValue):** 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.

13.92 Preset Value Register for Default Speed (presetvalue1)—Offset 62h

This register is used to read the SDCLK Frequency Select Value,Clock Generator Select Value,Driver Strength Select Value for Default Speed

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 4h

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RO	**DriverStrengthSelectValue (DriverStrengthSelectValue):** Driver Strength Select Value
13:11	0h RO	Reserved.
10	0h RO	**ClockGeneratorSelectValue (ClockGeneratorSelectValue):** This bit is effective when Host Controller supports programmable clock

Bit Range	Default & Access	Field Name (ID): Description
9:0	4h RO	**SDCLKFrequencySelectValue (SDCLKFrequencySelectValue):** 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.

13.93 Preset Value Register for High Speed (presetvalue2)—Offset 64h

This register is used to read the SDCLK Frequency Select Value,Clock Generator Select Value,Driver Strength Select Value for High Speed

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 2h

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RO	**DriverStrengthSelectValue (DriverStrengthSelectValue):** Driver Strength Select Value
13:11	0h RO	Reserved.
10	0h RO	**ClockGeneratorSelectValue (ClockGeneratorSelectValue):** This bit is effective when Host Controller supports programmable clock
9:0	2h RO	**SDCLKFrequencySelectValue (SDCLKFrequencySelectValue):** 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.

13.94 Preset Value Register for SDR12 (presetvalue3)—Offset 66h

This register is used to read the SDCLK Frequency Select Value,Clock Generator Select Value,Driver Strength Select Value for SDR12

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 4h

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RO	**DriverStrengthSelectValue (DriverStrengthSelectValue):** Driver Strength Select Value

Bit Range	Default & Access	Field Name (ID): Description
13:11	0h RO	Reserved.
10	0h RO	**ClockGeneratorSelectValue (ClockGeneratorSelectValue):** This bit is effective when Host Controller supports programmable clock
9:0	4h RO	**SDCLKFrequencySelectValue (SDCLKFrequencySelectValue):** 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.

13.95 Preset Value Register for SDR25 (presetvalue4)—Offset 68h

This register is used to read the SDCLK Frequency Select Value,Clock Generator Select Value,Driver Strength Select Value SDR25

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 2h

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RO	**DriverStrengthSelectValue (DriverStrengthSelectValue):** Driver Strength Select Value
13:11	0h RO	Reserved.
10	0h RO	**ClockGeneratorSelectValue (ClockGeneratorSelectValue):** This bit is effective when Host Controller supports programmable clock
9:0	2h RO	**SDCLKFrequencySelectValue (SDCLKFrequencySelectValue):** 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.

13.96 Preset Value Register for SDR50 (presetvalue5)—Offset 6Ah

This register is used to read the SDCLK Frequency Select Value,Clock Generator Select Value,Driver Strength Select Value for SDR50

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RO	**DriverStrengthSelectValue (DriverStrengthSelectValue):** Driver Strength Select Value
13:11	0h RO	Reserved.
10	0h RO	**ClockGeneratorSelectValue (ClockGeneratorSelectValue):** This bit is effective when Host Controller supports programmable clock
9:0	1h RO	**SDCLKFrequencySelectValue (SDCLKFrequencySelectValue):** 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.

13.97 Preset Value Register for SDR104 (presetvalue6)—Offset 6Ch

This register is used to read the SDCLK Frequency Select Value,Clock Generator Select Value,Driver Strength Select Value for SDR 104

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RO	**DriverStrengthSelectValue (DriverStrengthSelectValue):** Driver Strength Select Value
13:11	0h RO	Reserved.
10	0h RO	**ClockGeneratorSelectValue (ClockGeneratorSelectValue):** This bit is effective when Host Controller supports programmable clock
9:0	0h RO	**SDCLKFrequencySelectValue (SDCLKFrequencySelectValue):** 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.

13.98 Preset Value Register for DDR50 (presetvalue7)—Offset 6Eh

This register is used to read the SDCLK Frequency Select Value,Clock Generator Select Value,Driver Strength Select Value for DDR50

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 2h

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RO	**DriverStrengthSelectValue (DriverStrengthSelectValue):** Driver Strength Select Value
13:11	0h RO	Reserved.
10	0h RO	**ClockGeneratorSelectValue (ClockGeneratorSelectValue):** This bit is effective when Host Controller supports programmable clock
9:0	2h RO	**SDCLKFrequencySelectValue (SDCLKFrequencySelectValue):** 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.

13.99 Boot Timeout Control Register (boottimeoutcnt)—Offset 70h

This is used to program the boot timeout value counter

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**boot_timeoutcnt (boot_timeoutcnt):** This value determines the interval by which DAT line time-outs are detected during boot operation for eMMC4.4 card.

13.100 Preset Value Register for DDR50 (presetvalue8)—Offset 74h

This register is used to read the SDCLK Frequency Select Value,Clock Generator Select Value,Driver Strength Select Value for DDR50

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RO	**DriverStrengthSelectValue (DriverStrengthSelectValue):** Driver Strength Select Value

Bit Range	Default & Access	Field Name (ID): Description
13:11	0h RO	Reserved.
10	0h RO	**ClockGeneratorSelectValue (ClockGeneratorSelectValue):** This bit is effective when Host Controller supports programmable clock
9:0	0h RO	**SDCLKFrequencySelectValue (SDCLKFrequencySelectValue):** 10-bit preset value to set SDCLK Frequency Select in the Clock Control Register is described by a host system.

13.101 Vendor Register (vendreg)—Offset 78h

Vendor Register

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:1	0h RO	**RSVD (Reserved_15_1):** Reserved_15_1
0	0h RW	**The bit enables the enhanced strobe logic of the Host Controller (vendor_enhancedstrobe):** vendor_enhancedstrobe

13.102 Slot Interrupt Status Register (slotintrsts)—Offset FCh

This register is used to read the interrupt signal for each slot.

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	Reserved.
7:0	0h RO	**sdhchostif_slotintrsts (sdhchostif_slotintrsts):** These status bits indicate the logical OR of Interrupt signal and Wakeup signal for each slot.

13.103 Host Controller Version Register (hostcontrollerver)—Offset FEh

This register is used to read the vendor version number and specification version number

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 1002h

Bit Range	Default & Access	Field Name (ID): Description
15:8	10h RO	**SDHC_VENVERNUM (SDHC_VENVERNUM):** This status is reserved for the vendor version number.
7:0	2h RO	**SpecificationVersionNumber (SpecificationVersionNumber):** This status indicates the Host Controller Spec. Version. The upper and lower 4-bits indicate the version.

13.104 CQVER (CQVER)—Offset 200h

This register provides information about the version of the eMMC CQ standard which is 285 implemented by the CQE, in BCD format

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 510h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**RSVD (Reserved_51):** Reserved_51
11:8	5h RO	**eMMC Major Version Number (eMMC_Major_Version_Number):** eMMC Major Version Number (digit left of decimal point), in BCD format
7:4	1h RO	**eMMC Minor Version Number (eMMC_Minor_Version_Number):** eMMC Minor Version Number (digit right of decimal point), in BCD format
3:0	0h RO	**eMMC Version Suffix (eMMC_Version_Suffix):** eMMC Version Suffix (2nd digit right of decimal point), in BCD format

13.105 CQCAP (CQCAP)—Offset 204h

Command Queueing Capabilities

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 30C0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RSVD (Reserved_52):** Reserved_52
15:12	3h RO	**Internal Timer Clock Frequency Multiplier (Internal_Timer_Clock_Frequency_Multiplier):** ITCFMUL and ITCFVAL indicate the frequency of the clock used for interrupt coalescing timer and for determining the SQS polling period. See ITCFVAL definition for details.
11:10	0h RO	**RSVD (Reserved_53):** Reserved_53
9:0	C0h RO	**Internal Timer Clock Frequency Value (Internal_Timer_Clock_Frequency_Value):** TCFMUL and ITCFVAL indicate the frequency of the clock used for interrupt coalescing timer and for determining the polling period when using periodic SEND_QUEUE_ STATUS (CMD13) polling.

13.106 CQCFG (CQCFG)—Offset 208h

Command Queueing Configuration

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:13	0h RO	**RSVD (Reserved_54):** Reserved_54
12	0h RW	**Direct Command Enable (Direct_Command_Enable):** This bit indicates to the hardware whether the Task Descriptor in slot #31 of the TDL is a Data Transfer Task Descriptor, or a Direct Command Task Descriptor.
11:9	0h RO	**RSVD (Reserved_11_9):** Reserved_11_9
8	0h RW	**Task Descriptor Size (Task_Descriptor_Size):** This bit indicates whether the task descriptor size is 128 bits or 64 bits. This bit can only be configured when Command Queueing Enable bit is ?0? (command queueing is disabled)
7:1	0h RO	**RSVD (Reserved_7_1):** Reserved_7_1

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**Command Queueing Enable (Command_Queueing_Enable):** Software shall write ?1? this bit when in order to enable command queueing mode (i.e. enable CQE).

13.107 CQCTL (CQCTL)—Offset 20Ch

Command Queueing Control

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**RSVD (Reserved_57):** Reserved_57
8	0h RW	**Clear_All_Tasks (Clear_All_Tasks):** Software shall write 1 this bit when it wants to clear all the tasks sent to the device.
7:1	0h RO	**RSVD (Reserved_58):** Reserved_58
0	0h RW	**Halt (Halt):** Host software shall write 1 to the bit when it wants to acquire software control over the eMMC bus and disable CQE from issuing commands on the bus.

13.108 Command Queueing Interrupt Status (CQIS)—Offset 210h

This register indicates pending interrupts that require service. Each bit in this registers is asserted 296 in response a specific event, only if the respective bit is set in CQISTE register.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**RSVD (Reserved_59):** Reserved_59
3	0h RW/1C	**Task_Cleared (Task_Cleared):** Task_ClearedThis status bit is asserted (if CQISTE.TCL=1) when a task clear operation is completed by CQE. The completed task clear operation is either an individual task clear (CQTCLR) or clearing of all tasks (CQCTL).

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW/1C	**Response_Error_Detected_Interrupt (Response_Error_Detected_Interrupt):** This status bit is asserted (if CQISTE.RED=1) when a response is received with an error bit set in the device status field.
1	0h RW/1C	**Task_Complete_Interrupt (Task_Complete_Interrupt):** This status bit is asserted (if CQISTE.TCC=1) when at least one of the following two conditions are met: (1) A task is completed and the INT bit is set in its Task Descriptor (2) Interrupt caused by Interrupt Coalescing logic
0	0h RW/1C	**Halt_Complete_Interrupt (Halt_Complete_Interrupt):** This status bit is asserted (if CQISTE.HAC=1) when halt bit in CQCTL register transitions from 0 to 1 indicating that host controller has completed its current ongoing task and has entered halt state.

13.109 Command Queueing Interrupt Status Enabled (CQISTE)—Offset 214h

This register enables and disables the reporting of the corresponding interrupt to host software in 299 CQIS register

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**RSVD (Reserved_60):** Reserved_60
3	0h RW	**Task_Cleared_Status_Enable (Task_Cleared_Status_Enable):** 1 = CQIS.TCL will be set when its interrupt condition is active 0 = CQIS.TCL is disabled
2	0h RW	**Response_Error_Detected_Status_Enable (Response_Error_Detected_Status_Enable):** 1 = CQIS.RED will be set when its interrupt condition is active 0 = CQIS.RED is disabled
1	0h RW	**Task_Complete_Status_Enable (Task_Complete_Status_Enable):** 1 = CQIS.TCC will be set when its interrupt condition is active 0 = CQIS.TCC is disabled
0	0h RW	**Halt_Complete_Status_Enable (Halt_Complete_Status_Enable):** 1 = CQIS.HAC will be set when its interrupt condition is active 0 = CQIS.HAC is disabled

13.110 Command Queueing Interrupt Status Enable (CQISGE)—Offset 218h

This register enables and disables the generation of interrupts to host software

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (Reserved_61):** Reserved_61
3	0h RW	**Task_Cleared_Signal_Enable (Task_Cleared_Signal_Enable):** When set and CQIS.TCL is asserted, the CQE shall generate an interrupt
2	0h RW	**Response_Error_Detected_Signal_Enable (Response_Error_Detected_Signal_Enable):** When set and CQIS.RED is asserted, the CQE shall generate an interrupt
1	0h RW	**Task_Complete_Signal_Enable (Task_Complete_Signal_Enable):** When set and CQIS.TCC is asserted, the CQE shall generate an interrupt
0	0h RW	**Halt_Complete_Signal_Enable (Halt_Complete_Signal_Enable):** When set and CQIS.HAC is asserted, the CQE shall generate an interrupt

13.111 Interrupt Coalescing (CQIC)—Offset 21Ch

This register controls the interrupt coalescing feature.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**intrcoalescreg_enb (intrcoalescreg_enb):** When set to 0 by software, command responses are neither counted nor timed.When set to 1, the interrupt coalescing mechanism is enabled and coalesced interrupts are generated.
30:21	0h RO	**Reserved (Reserved_62):** Reserved_62

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**intrcoalescreg_icsb (intrcoalescreg_icsb):** This bit indicates to software whether any tasks (with INT=0) have completed and counted towards interrupt coalescing
19:17	0h RO	**Reserved (Reserved_63):** Reserved_63
16	0h WO	**intrcoalescreg_icctr (intrcoalescreg_icctr):** When host driver writes 1, the interrupt coalescing timer and counter are reset
15	0h WO	**intrcoalescreg_iccthwen (intrcoalescreg_iccthwen):** When software writes 1, the value ICCTH is updated with the contents written at the same cycle. When software writes 0, the value in ICCTH is not updated.
14:13	0h RO	**Reserved (Reserved_64):** Reserved_64
12:8	0h RW	**intrcoalescreg_iccth (intrcoalescreg_iccth):** Software uses this field to configure the number of task completions (only tasks with INT=0 in the Task Descriptor) which are required in order to generate an interrupt.
7	0h WO	**intrcoalescreg_ictovalwen (intrcoalescreg_ictovalwen):** When software writes 1, the value ICTOVAL is updated with the contents written at the same cycle. When software writes 0, the value in ICTOVAL is not updated.
6:0	0h RW	**intrcoalescreg_ictoval (intrcoalescreg_ictoval):** Software uses this field to configure the maximum time allowed between the completion of a task on the bus and the generation of an interrupt.

13.112 Command Queueing Task Descriptor List Base Address (CQTDLBA)—Offset 220h

This register is used for configuring the lower 32 bits of the byte address of the head of the Task 312 Descriptor List in the host memory.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**cmdquetaskdesclistbaseaddrreg_tdlba (cmdquetaskdesclistbaseaddrreg_tdlba):** This register stores the LSB bits (bits 31:0) of the byte address of the head of the Task Descriptor List in system memory.

13.113 Command Queueing Task Descriptor List Base Address Upper 32 Bits (CQTDLBAU)—Offset 224h

This register is used for configuring the upper 32 bits of the byte address of the head of the Task 316 Descriptor List in the host memory.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**cmdquetaskdesclistbaseadduprreg_tdlba (cmdquetaskdesclistbaseadduprreg_tdlba):** This register stores the MSB bits (bits 63:32) of the byte address of the head of the Task Descriptor List in system memory.

13.114 Command Queueing Taks Doorbell (CQTDRB)—Offset 228h

Using this register, software triggers CQE to process a new task.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**cmdquetaskdoorbellreg (cmdquetaskdoorbellreg):** Writing 1 to bit n of this register triggers CQE to start processing the task encoded in slot n of the TDL.

13.115 Command Queueing Taks Doorbell Notification (CQTCN)—Offset 22Ch

This register is used by CQE to notify software about completed tasks.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**cmdquetaskdoorbellnotifyreg_ntf (cmdquetaskdoorbellnotifyreg_ntf):** When receiving interrupt for task completion, software may read this register to know which tasks have finished. After reading this register, software may clear the relevant bit fields by writing 1 to the corresponding bits.

13.116 Command Queueing Device Queue Status (CQDQS)—Offset 230h

This register stores the most recent value of the device?s queue status.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**cmdquedevquestsreg_sts (cmdquedevquestsreg_sts):** Every time the Host controller receives a queue status register (QSR) from the device, it updates this register with the response of status command, i.e. the device?s queue status.

13.117 Command Queueing Device Pending Tasks (CQDPT)—Offset 234h

This register indicates to software which tasks are queued in the device, awaiting execution.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**cmdquedevpendtaskreg_pend (cmdquedevpendtaskreg_pend):** CQE shall set this bit after receiving a successful response for CMD45. CQE shall clear this bit after the task has completed execution.

13.118 Command Queueing Task Clear (CQTCLR)—Offset 238h

This register is used for removing an outstanding task in the CQE. 327. The register should be used only when CQE is in Halt state.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**cmdquequetaskclrreg_clr (cmdquequetaskclrreg_clr):** Writing 1 to bit n of this register orders CQE to clear a task which software has previously issued.

13.119 Send Status Timer Configuration 1 (CQSSC1)—Offset 240h

The register controls the when SEND_QUEUE_STATUS commands are sent.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 11000h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RO	**Reserved (Reserved_65):** Reserved_65
19:16	1h RW	**sendststmrconfig1reg_blkcnt (sendststmrconfig1reg_blkcnt):** This field indicates to CQE when to send SEND_QUEUE_STATUS (CMD13) command to inquire the status of the device's task queue.
15:0	1000h RW	**sendststmrconfig1reg_idltmr (sendststmrconfig1reg_idltmr):** This field indicates to CQE the polling period to use when using periodic SEND_QUEUE_STATUS (CMD13) polling.

13.120 Send Status Configuration 2 (CQSSC2)—Offset 244h

This register is used for 333 configuring RCA field in SEND_QUEUE_STATUS command argument.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (Reserved_66):** Reserved_66
15:0	0h RW	**sendstsconfigreg_rca (sendstsconfigreg_rca):** This field provides CQE with the contents of the 16-bit RCA field in SEND_QUEUE_ STATUS (CMD13) command argument.

13.121 Command Response for Direct-Command Task (CQCRDCT)—Offset 248h

This register is used for passing the response of a DCMD task to software.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**cmdrespfordirectcmndtaskreg_lstresp (cmdrespfordirectcmndtaskreg_lstresp):** This register contains the response of the command generated by the last direct-command (DCMD) task which was sent.

13.122 Response Mode Error Mask (CQRMEM)—Offset 250h

This register controls the generation of Response Error Detection (RED) interrupt.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: FDF9A080h

Bit Range	Default & Access	Field Name (ID): Description
31:0	FDF9A080h RW	**resmodeerrmaskreg_errmask (resmodeerrmaskreg_errmask):** This bit is used as in interrupt mask on the device status filed which is received in R1/R1b responses.

13.123 CQBASE+54h:CQTERRI - Task Error Information (CQTERRI)—Offset 254h

This register is updated by CQE when an error occurs on data or command related to a task activity. When such error is detected by CQE or indicated by the eMMC controller CQE stores in CQTERRI the task IDs and the command indices of the commands which were executed on the 343 command line and data lines when the error occurred.

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**cqtaskerrinforreg_valid (cqtaskerrinforreg_valid):** This bit is updated when an error is detected by CQE, or indicated by eMMC controller.
30:29	0h RO	**Reserved (Reserved_67):** Reserved_67
28:24	0h RO	**cqtaskerrinforreg_tskID (cqtaskerrinforreg_tskID):** This field indicates the ID of the task which was executed on the data lines when an error occurred.
23:22	0h RO	**Reserved (Reserved_68):** Reserved_68
21:16	0h RO	**cqtaskerrinforreg_index (cqtaskerrinforreg_index):** This field indicates the index of the command which was executed on the data lines when an error occurred.
15	0h RO	**cqtaskerrinforreg_rspmode_errvalid (cqtaskerrinforreg_rspmode_errvalid):** This bit is updated when an error is detected by CQE, or indicated by eMMC controller.
14:13	0h RO	**Reserved (Reserved_69):** Reserved_69
12:8	0h RO	**cqtaskerrinforreg_resmodeerrtaskID (cqtaskerrinforreg_resmodeerrtaskID):** This field indicates the ID of the task which was executed on the command line when an error occurred.
7:6	0h RO	**Reserved (Reserved_70):** Reserved_70
5:0	0h RO	**cqtaskerrinforreg_resmoderrcmdindex (cqtaskerrinforreg_resmoderrcmdindex):** This field indicates the index of the command which was executed on the command line when an error occurred.

13.124 Command Response Index (CQCRI)—Offset 258h

This register stores the index of the last received command response.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RO	**Reserved (Reserved_71):** Reserved_71
5:0	0h RO	**cmdrespindexreg_lstcmdresindex (cmdrespindexreg_lstcmdresindex):** This field stores the index of the last received command response. CQE shall update the value every time a command response is received.

13.125 Command Response Argument (CQCRA)—Offset 25Ch

This register stores the index of the last received command response.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**cmdrespargument_lstcmdresarg (cmdrespargument_lstcmdresarg):** This field stores the argument of the last received command. CQE shall update the value every time a command response is received.

§ §

14 LPIO Registers

This chapter documents the registers of the LPIO controller devices. The processor contains multiple LPIO controller devices:

- Bus: 0, Device: 22, Function: 0 (LPIO0)
- Bus: 0, Device: 22, Function: 1 (LPIO1)
- Bus: 0, Device: 22, Function: 2 (LPIO2)
- Bus: 0, Device: 22, Function: 3 (LPIO3)
- Bus: 0, Device: 23, Function: 0 (LPIO4)
- Bus: 0, Device: 23, Function: 1 (LPIO5)
- Bus: 0, Device: 23, Function: 2 (LPIO6)
- Bus: 0, Device: 23, Function: 3 (LPIO7)
- Bus: 0, Device: 24, Function: 0 (LPIO8)
- Bus: 0, Device: 24, Function: 1 (LPIO9)
- Bus: 0, Device: 24, Function: 2 (LPIO10)
- Bus: 0, Device: 24, Function: 3 (LPIO11)
- Bus: 0, Device: 25, Function: 0 (LPIO12)
- Bus: 0, Device: 25, Function: 1 (LPIO13)
- Bus: 0, Device: 25, Function: 2 (LPIO14)

NOTE: Register default values are taken from device LPIO0 only. Refer Vol1 for Device IDs

14.1 DEVICEVENDORID - Device ID and Vendor ID Register (DEVVENDID) — Offset 0h

Device ID and Vendor ID provided by this register uniquely identifies the particular Device

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + 0h	AAC8086 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	AACh RO	**DEVICEID:** Device ID identifies the particular PCI device
15:0	8086h RO	**VENDORID:** Vendor ID is a unique ID provided by the PCI SIG, which identifies the manufacturer of the device

14.2 STATUSCOMMAND — Offset 4h

Command register to programme interrupt disable , bus master enable, and Memory space enable. Status register to read the errors and aborts

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + 4h	100000 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**RESERVED0:** Reserved
29	0h RW/1C	**RMA:** Received Master Abort
28	0h RW/1C	**RTA:** Received Target Abort
27:21	0h RO	**RESERVED1:** Reserved
20	1h RO	**CAPLIST:** Capabilities List: Indicates that the controller contains a capabilities pointer list
19	0h RO	**INTR_STATUS:** Interrupt Status: This bit reflects state of interrupt in the device
18:16	0h RO	**RESERVED2:** Reserved
15:11	0h RO	**RESERVED3:** Reserved
10	0h RW	**INTR_DISABLE:** Interrupt Disable
9	0h RO	**RESERVED4:** Reserved
8	0h RW	**SERR_ENABLE:** SERR Enable , Not implemented
7:3	0h RO	**RESERVED5:** Reserved
2	0h RW	**BME:** Bus Master Enable
1	0h RW	**MSE:** Memory Space Enable
0	0h RO	**RESERVED6:** Reserved

14.3 REVCLASSCODE — Offset 8h

Revision ID register identifies revision of particular device and Class Code register is used to identify generic function of the device

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + 8h	11800000 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	118000h RO	**CLASS_CODES:** Class Code register is read-only and is used to identify the generic function of the device, and in some cases, a specific register-level programming interface
7:0	0h RO	**RID:** Revision ID identifies the revision of particular PCI device.

14.4 CLLATHEADERBIST — Offset Ch

Cache Line size as RW with def 0, Latency timer RW with def 0, Header type with Type 0 configuration header and Reserved BIST register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + Ch	800000 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**RESERVED0:** Reserved
23	1h RO	**MULFNDEV:** Multi-Function Device
22:16	0h RO	**HEADERTYPE:** Header Type: Implements Type 0 Configuration header
15:8	0h RO	**LATTIMER:** Latency Timer:. This register is implemented as R/W with default as 0
7:0	0h RW	**CACHELINE_SIZE:** Cacheline Size

14.5 BAR — Offset 10h

Base Address Register low [31:2] , type[2:1] in 32bit or 64bit addr range and memory space indicator [0]

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + 10h	4 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**BASEADDR:** Base Address Register Low, Base address of the OCP fabric memory space. Taken from Strap values as ones
11:4	0h RO	**SIZEINDICATOR:** Size Indicator RO Always returns 0 The size of this register depends on the size of the memory space
3	0h RO	**PREFETCHABLE:** Prefetchable: Indicates that this BAR is not prefetchable
2:1	2h RO	**TYPE:** If BAR_64b_EN is 0 then 00 indicates BAR lies in 32bit address range, If BAR_64b_EN is 1 then 10 Indicates BAR lies in 64 bit address range
0	0h RO	**MESSAGE_SPACE:** Memory Space Indicator: 0 indicates this BAR is present in the memory space.

14.6 BAR -Base Address Register High (BAR_HIGH) — Offset 14h

Base Address Register High enabled if [2:1] of BAR_type_LOW is 10

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + 14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**BASEADDR_HIGH:** Base Address high - MSB

14.7 BAR1 — Offset 18h

Base Address Register1 accesses to PCI configuration space and is always 4K, type in [2:1] and memory space indicator in [0]

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + 18h	4 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**BASEADDR1:** Base Address1 This field is present if BAR1 is enabled through private configuration space.
11:4	0h RO	**SIZEINDICATOR1:** Always is 0 as minimum size is 4K
3	0h RO	**PREFETCHABLE1:** Prefetchable: Indicates that this BAR is not prefetchable.
2:1	2h RO	**TYPE1:** If BAR_64b_EN is 0 then 00 indicates BAR lies in 32bit address range If BAR_64b_EN is 1 then 10 Indicates BAR lies in 64 bit address range

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**MESSAGE_SPACE1:** Memory Space Indicator: 0 Indicates this BAR is present in the memory space

14.8 BAR1 -Base Address Register1 High (BAR1_HIGH) — Offset 1Ch

Base Address Register1 High enabled only if [2:1] of BAR1 register is 10

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + 1Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**BASEADDR1_HIGH:** Base Address: Base address of the OCP fabric memory space. Taken from Strap values as ones

14.9 SUBSYSTEMID — Offset 2Ch

SVID register along with SID register is to distinguish subsystem from another

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**SUBSYSTEMID:** Subsystem ID: This register is implemented for any function that can be instantiated more than once in a given system.
15:0	0h RW/O	**SUBSYSTEMVENDORID:** Subsystem Vendor ID: This register must be implemented for any function that can be instantiated more than once in a given system

14.10 EXPANSION ROM base address (EXPANSION_ROM_BASEADDR) — Offset 30h

EXPANSION ROM base address register is a RO indicates support for expansion ROMs

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + 30h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**EXPANSION_ROM_BASE:** Expansion ROM Base Address: Value of all zeros indicates no support for Expansion ROM

14.11 CAPABILITYPTR — Offset 34h

Capabilities Pointer register indicates what the next capability is

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + 34h	80 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**RESERVED0:** Reserved
7:0	80h RO	**CAPPTR_POWER:** Capabilities Pointer: Indicates what the next capability is.

14.12 INTERRUPTREG — Offset 3Ch

Interrupt line Register isn't used in Bridge directly, Interrupt Pin register reflects the IPIN value in private config space. Min_gnt register indicating the req of latency timers and max_lat register max latency

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + 3Ch	100 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**MAX_LAT:** Value of 0 indicates device has no major requirements for the settings of latency timers
23:16	0h RO	**MIN_GNT:** Value of 0 indicates device has no major requirements for the settings of latency timers
15:12	0h RO	**RESERVED0:** Reserved
11:8	1h RO	**INTPIN:** Interrupt Pin: Value in this register is reflected from the IPIN value in the private configuration space.
7:0	0h RW	**INTLINE:** Interrupt Line: It is used to communicate to software, the interrupt line to which the interrupt pin is connected

14.13 POWERCAPID — Offset 80h

PowerManagement Capability ID register points to next capability structure and power mgmnt capability , with Power management capabilities register for PME support and version

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + 80h	39001 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	0h RO	**PMESUPPORT:** This 5-bit field indicates the power states in which the function can assert the PME#
26:19	0h RO	**RESERVED0:** Reserved
18:16	3h RO	**VERSION:** Indicates support for Revision 1.2 of the PCI Power Management Specification
15:8	90h RO	**NXTCAP:** Next Capability: Points to the next capability structure.
7:0	1h RO	**POWER_CAP:** Power Management Capability: Indicates this is power management capability

14.14 PMECTRLSTATUS — Offset 84h

power management control and status register to set and read PME status, PME enable, No Soft reset and power state

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + 84h	8 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED0:** Reserved
15	0h RW/1C	**PMESTATUS:** PME Status
14:9	0h RO	**RESERVED1:** Reserved
8	0h RW	**PMEENABLE:** PME Enable
7:4	0h RO	**RESERVED2:** Reserved
3	1h RO	**NO_SOFT_RESET:** This bit indicates that devices transitioning from D3hot to D0 because of Powerstate commands do not perform an internal reset

Bit Range	Default & Access	Field Name (ID): Description
2	0h RO	**RESERVED3:** Reserved
1:0	0h RW	**POWERSTATE:** Power State: This field is used both to determine the current power state and to set a new power state

14.15 PCI DEVICE IDLE CAPABILITY RECORD (PCIDEVIDLE_CAP_RECORD) — Offset 90h

PCI Device Vendor Specific Capability register defines Vendor specific Capability ID, revision , length , next capability and CAPID

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + 90h	F0140009 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RO	**VEND_CAP:** Vendor Specific Capability ID
27:24	0h RO	**REVID:** Revision ID of capability structure
23:16	14h RO	**CAP_LENGTH:** Vendor Specific Capability Length
15:8	0h RO	**NEXT_CAP:** Next Capability
7:0	9h RO	**CAPID:** Capability ID

14.16 DEVID VENDOR SPECIFIC REG (DEVID_VEND_SPECIFIC_REG) — Offset 94h

Extended Vendor capability register for VSEC Length, revision and ID

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + 94h	1400010 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	14h RO	**VSEC_LENGTH:** Vendor Specific Extended Capability Length
19:16	0h RO	**VSEC_REV:** Vendor specific Extended Capability revision
15:0	10h RO	**VSECID:** Vendor Specific Extended Capability ID

14.17 D0I3_CONTROL_SW_LTR_MMIO_REG — Offset 98h

Software location pointer in MMIO space as an offset specified by BAR

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + 98h	2101 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	210h RO	**SW_LAT_DWORD_OFFSET:** SW LTR Update MMIO Offset Location (SWLTRLOC)
3:1	0h RO	**SW_LAT_BAR_NUM:** Indicates that the SW LTR update MMIO location is always at BAR0
0	1h RO	**SW_LAT_VALID:** This value is reflected from the SW LTR valid strap at the top level

14.18 DEVICE_IDLE_POINTER_REG — Offset 9Ch

Device IDLE pointer register giving details on Device MMIO offset location , BAR NUM and D0i3 Valid Strap

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + 9Ch	24C1 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	24Ch RO	**DWORD_OFFSET:** contains the location pointer to the SW LTR register in MMIO space, as an offset from the specified BAR
3:1	0h RO	**BAR_NUM:** Bar num: Indicates that the D0i3 MMIO location is always at BAR0
0	1h RO	**VALID:** Valid: This value is reflected from the D0i3 valid strap at the top level.

14.19 D0I3_MAX_POW_LAT_PG_CONFIG — Offset A0h

D0idle_Max_Power_On_Latency register set at boot and Power control enable register to enable communication with the PGCB block below the Bridge

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + A0h	F0800 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**RESERVED0:** Reserved
21	0h RO	**HAE:** Hardware Autonomous Enable - LPSS has RO instead of RW
20	0h RO	**RESERVED1:** Reserved
19	1h RW	**SLEEP_EN:** Sleep Enable
18	1h RW	**PGE:** DEVIDLE Enable (DEVIDLEN): If ?1?, then the function will power gate when idle and the DevIdle register (DevIdleC[2] = ?1?) is set.
17	1h RW	**I3_ENABLE:** D3-Hot Enable (D3HEN): If ?1?, then function will power gate when idle and the PMCSR[1:0] register in the function =?11? (D3).
16	1h RW	**PMCRE:** PMCRE: PMC Request Enable
15:13	0h RO	**RESERVED2:** Reserved
12:10	2h RW/O	**POW_LAT_SCALE:** Power On Latency Scale
9:0	0h RW/O	**POW_LAT_VALUE:** Power On Latency value

14.20 GEN_REGRW1 - General Purpose Read Write Register1 (GEN_REGRW1) — Offset B0h

General Purpose PCI Read Write Register1

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + B0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW1:** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg1 Out of Band signal

14.21 GEN_REGRW2 — Offset B4h

General Purpose PCI Read Write Register2

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + B4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW2:** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg2 Out of Band signal

14.22 GEN_REGRW3 — Offset B8h

General Purpose PCI Read Write Register3

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + B8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW3:** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg3 Out of Band signal

14.23 GEN_REGRW4 — Offset BCh

General Purpose PCI Read Write Register4

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + BCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW4:** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg4 Out of Band signal

14.24 GEN_INPUT_REG — Offset C0h

General Purpose Input Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + C0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**GEN_REG_INPUT_RW:** General Purpose Input Register: This register value reflects the value of oob_gen_input_pci Out of Band signal

14.25 MANID — Offset F8h

Manufacturers ID register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:22, F:0] + F8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**MANID:** Manufacturer ID: Default value comes from straps.

§ §

15 LPIO BAR Registers

This chapter documents the LPIOBAR registers. Base address of these registers are defined in the BAR register of each LPIO device. The processor contains multiple LPIO controller devices:

- Bus: 0, Device: 22, Function: 0 (LPIO0)
- Bus: 0, Device: 22, Function: 1 (LPIO1)
- Bus: 0, Device: 22, Function: 2 (LPIO2)
- Bus: 0, Device: 22, Function: 3 (LPIO3)
- Bus: 0, Device: 23, Function: 0 (LPIO4)
- Bus: 0, Device: 23, Function: 1 (LPIO5)
- Bus: 0, Device: 23, Function: 2 (LPIO6)
- Bus: 0, Device: 23, Function: 3 (LPIO7)
- Bus: 0, Device: 24, Function: 0 (LPIO8)
- Bus: 0, Device: 24, Function: 1 (LPIO9)
- Bus: 0, Device: 24, Function: 2 (LPIO10)
- Bus: 0, Device: 24, Function: 3 (LPIO11)
- Bus: 0, Device: 25, Function: 0 (LPIO12)
- Bus: 0, Device: 25, Function: 1 (LPIO13)
- Bus: 0, Device: 25, Function: 2 (LPIO14)

NOTE: Register default values are taken from device LPIO0 only. Refer Vol1 for Device IDs

15.1 IC_CON — Offset 0h

I2C Control Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 0h	77 h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h NA	**Reserved (RESERVED_31_9):** Reserved
8	0h RW	**TX_EMPTY_CTRL:**
7	0h NA	**RESERVED_7:** Reserved

Bit Range	Default & Access	Field Name (ID): Description
6	1h RW	**IC_SLAVE_DISABLE:**
5	1h RW	**IC_RESTART_EN:**
4	1h RO	**IC_10BITADDR_MASTER (IC_10BITADDR_MASTER_RD_ONLY):**
3	0h NA	**RESERVED_3:** Reserved
2:1	3h RW	**SPEED:**
0	1h RW	**MASTER_MODE:**

15.2 IC_TAR — Offset 4h

I2C Target Address Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 4h	1055 h

Bit Range	Default & Access	Field Name (ID): Description
31:13	0h NA	**RESERVED_13_31:** Reserved
12	1h RW	**IC_10BITADDR_MASTER:**
11	0h RW	**SPECIAL:**
10	0h RW	**GC_OR_START:**
9:0	55h RW	**IC_TAR:**

15.3 IC_SAR — Offset 8h

I2C Slave Address Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 8h	55 h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h NA	**RESERVED_10_31:** Reserved
9:0	55h RW	**IC_SAR:**

15.4 IC_HS_MADDR — Offset Ch

I2C High Speed Master Mode Code Address Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + Ch	1 h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h NA	**RESERVED_3_31:** Reserved
2:0	1h RW	**IC_HS_MAR:**

15.5 IC_DATA_CMD — Offset 10h

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 10h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h NA	**RESERVED_11_31:** Reserved
10	0h RW	**RESTART:**
9	0h RW	**STOP:**
8	0h RW	**CMD:**
7:0	0h RW	**DAT:**

15.6 IC_SS_SCL_HCNT — Offset 14h

Standard Speed I2C Clock SCL High Count Register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 14h	1F4 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h NA	**RESERVED_16_31:** Reserved
15:0	1F4h RW	**IC_SS_SCL_HCNT:**

15.7 IC_SS_SCL_LCNT — Offset 18h

Standard Speed I2C Clock SCL Low Count Register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 18h	24C h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h NA	**RESERVED_16_31:** Reserved
15:0	24Ch RW	**IC_SS_SCL_LCNT:**

15.8 IC_FS_SCL_HCNT — Offset 1Ch

Fast Speed I2C Clock SCL High Count Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 1Ch	4B h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h NA	**RESERVED_16_31:** Reserved
15:0	4Bh RW	**IC_FS_SCL_HCNT:**

15.9 IC_FS_SCL_LCNT — Offset 20h

Fast Speed I2C Clock SCL Low Count Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 20h	A3 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h NA	**RESERVED_16_31:** Reserved
15:0	A3h RW	**IC_FS_SCL_LCNT:**

15.10 IC_HS_SCL_HCNT — Offset 24h

High Speed I2C Clock SCL High Count Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 24h	8 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h NA	**RESERVED_16_31:** Reserved
15:0	8h RW	**IC_HS_SCL_HCNT:**

15.11 IC_HS_SCL_LCNT — Offset 28h

High Speed I2C Clock SCL Low Count Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 28h	14 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h NA	**RESERVED_16_31:** Reserved
15:0	14h RW	**IC_HS_SCL_LCNT:**

15.12 IC_INTR_STAT — Offset 2Ch

I2C Interrupt Status Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h NA	**RESERVED_14_31:** Reserved
13	0h RO	**R_MST_ON_HOLD:**
12	0h RO	**R_RESTART_DET:**
11	0h RO	**R_GEN_CALL:**
10	0h RO	**R_START_DET:**
9	0h RO	**R_STOP_DET:**
8	0h RO	**R_ACTIVITY:**
7	0h RO	**R_RX_DONE:**
6	0h RO	**R_TX_ABRT:**
5	0h RO	**R_RD_REQ:**
4	0h RO	**R_TX_EMPTY:**
3	0h RO	**R_TX_OVER:**
2	0h RO	**R_RX_FULL:**
1	0h RO	**R_RX_OVER:**
0	0h RO	**R_RX_UNDER:**

15.13 IC_INTR_MASK — Offset 30h

I2C Interrupt Mask Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 30h	8FF h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h NA	**RESERVED_31_14:** Reserved
13	0h RW	**M_MST_ON_HOLD:**
12	0h RO	**M_RESTART_DET:**
11	1h RW	**M_GEN_CALL:**
10	0h RW	**M_START_DET:**
9	0h RW	**M_STOP_DET:**
8	0h RW	**M_ACTIVITY:**
7	1h RW	**M_RX_DONE:**
6	1h RW	**M_TX_ABRT:**
5	1h RW	**M_RD_REQ:**
4	1h RW	**M_TX_EMPTY:**
3	1h RW	**M_TX_OVER:**
2	1h RW	**M_RX_FULL:**
1	1h RW	**M_RX_OVER:**
0	1h RW	**M_RX_UNDER:**

15.14 IC_RAW_INTR_STAT — Offset 34h

I2C Raw Interrupt Status Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 34h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h NA	**RESERVED_14_31:** Reserved
13	0h RO	**MST_ON_HOLD:** Same as in reg_IC_INTR_STAT

Bit Range	Default & Access	Field Name (ID): Description
12	0h RO	**RESTART_DET:** Same as in reg_IC_INTR_STAT
11	0h RO	**GEN_CALL:** Same as in reg_IC_INTR_STAT
10	0h RO	**START_DET:** Same as in reg_IC_INTR_STAT
9	0h RO	**STOP_DET:** Same as in reg_IC_INTR_STAT
8	0h RO	**ACTIVITY:** Same as in reg_IC_INTR_STAT
7	0h RO	**RX_DONE:** Same as in reg_IC_INTR_STAT
6	0h RO	**TX_ABRT:** Same as in reg_IC_INTR_STAT
5	0h RO	**RD_REQ:** Same as in reg_IC_INTR_STAT
4	0h RO	**TX_EMPTY:** Same as in reg_IC_INTR_STAT
3	0h RO	**TX_OVER:** Same as in reg_IC_INTR_STAT
2	0h RO	**RX_FULL:** Same as in reg_IC_INTR_STAT
1	0h RO	**RX_OVER:** Same as in reg_IC_INTR_STAT
0	0h RO	**RX_UNDER:** Same as in reg_IC_INTR_STAT

15.15 IC_RX_TL — Offset 38h

I2C Receive FIFO Threshold Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 38h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**RESERVED_8_31:** Reserved
7:0	0h RW	**RX_TL:**

15.16 IC_TX_TL — Offset 3Ch

I2C Transmit FIFO Threshold Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 3Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h NA	**RESERVED_8_31:** Reserved
7:0	0h RW	**TX_TL:**

15.17 IC_CLR_INTR — Offset 40h

Clear Combined and Individual Interrupt Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 40h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h NA	**RESERVED_1_31:** Reserved
0	0h RO	**CLR_INTR:**

15.18 IC_CLR_RX_UNDER — Offset 44h

Clear RX_UNDER Interrupt Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 44h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h NA	**RESERVED_1_31:** Reserved
0	0h RO	**CLR_RX_UNDER:**

15.19 IC_CLR_RX_OVER — Offset 48h

Clear RX_OVER Interrupt Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 48h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h NA	**RESERVED_1_31:** Reserved
0	0h RO	**CLR_RX_OVER:**

15.20 IC_CLR_TX_OVER — Offset 4Ch

Clear TX_OVER Interrupt Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 4Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h NA	**RESERVED_1_31:** Reserved
0	0h RO	**CLR_TX_OVER:** Read this register to clear the TX_OVER interrupt (bit 3) of the IC_RAW_INTR_STAT register.

15.21 IC_CLR_RD_REQ — Offset 50h

Clear RD_REQ Interrupt Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 50h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h NA	**RESERVED_1_31:** Reserved
0	0h RO	**CLR_RD_REQ:**

15.22 IC_CLR_TX_ABRT — Offset 54h

Clear TX_ABRT Interrupt Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 54h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h NA	**RESERVED_1_31:** Reserved
0	0h RO	**CLR_TX_ABRT:**

15.23 IC_CLR_RX_DONE — Offset 58h

Clear RX_DONE Interrupt Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 58h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h NA	**RESERVED_1_31:** Reserved
0	0h RO	**CLR_RX_DONE:**

15.24 IC_CLR_ACTIVITY — Offset 5Ch

Clear ACTIVITY Interrupt Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 5Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h NA	**RESERVED_1_31:** Reserved
0	0h RO	**CLR_ACTIVITY:**

15.25 IC_CLR_STOP_DET — Offset 60h

Clear STOP_DET Interrupt Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 60h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h NA	**RESERVED_1_31:** Reserved
0	0h RO	**CLR_STOP_DET:**

15.26 IC_CLR_START_DET — Offset 64h

Clear START_DET Interrupt Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 64h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h NA	**RESERVED_1_31:** Reserved
0	0h RO	**CLR_START_DET:**

15.27 IC_CLR_GEN_CALL — Offset 68h

Clear GEN_CALL Interrupt Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 68h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h NA	**RESERVED_1_31:** Reserved
0	0h RO	**CLR_GEN_CALL:**

15.28 IC_ENABLE — Offset 6Ch

I2C Enable Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 6Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h NA	**RESERVED_2_31:** Reserved
1	0h RW	**ABORT:**
0	0h RW	**ENABLE:**

15.29 IC_STATUS — Offset 70h

I2C Status Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 70h	6 h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h NA	**RESERVED_7_31:** Reserved
6	0h RO	**SLV_ACTIVITY:**
5	0h RO	**MST_ACTIVITY:**
4	0h RO	**RFF:**
3	0h RO	**RFNE:**
2	1h RO	**TFE:**
1	1h RO	**TFNF:**
0	0h RO	**ACTIVITY:** I2C Activity Status

15.30 IC_TXFLR — Offset 74h

I2C Transmit FIFO Level Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 74h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h NA	**RESERVED_9_31:** Reserved
8:0	0h RO	**TXFLR:**

15.31 IC_RXFLR — Offset 78h

I2C Receive FIFO Level Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 78h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h NA	**RESERVED_9_31:** Reserved
8:0	0h RO	**RXFLR:**

15.32 IC_SDA_HOLD — Offset 7Ch

I2C SDA Hold Time Length Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 7Ch	1 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h NA	**RESERVED_16_31:** Reserved
15:0	1h RW	**IC_SDA_HOLD:** Sets the required SDA hold time in units of ic_clk period.

15.33 IC_TX_ABRT_SOURCE — Offset 80h

I2C Transmit Abort Source Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 80h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**TX_FLUSH_CNT:**
23:17	0h NA	**RESERVED_23_17:** Reserved
16	0h RO	**ABRT_USER_ABRT:**
15	0h RO	**ABRT_SLVRD_INTX:**
14	0h RO	**ABRT_SLV_ARBLOST:**
13	0h RO	**ABRT_SLVFLUSH_TXFIFO:**
12	0h RO	**ARB_LOST:**
11	0h RO	**ABRT_MASTER_DIS:**
10	0h RO	**ABRT_10B_RD_NORSTRT:**
9	0h RO	**ABRT_SBYTE_NORSTRT:**
8	0h RO	**ABRT_HS_NORSTRT:**
7	0h RO	**ABRT_SBYTE_ACKDET:**
6	0h RO	**ABRT_HS_ACKDET:**
5	0h RO	**ABRT_GCALL_READ:**
4	0h RO	**ABRT_GCALL_NOACK:**
3	0h RO	**ABRT_TXDATA_NOACK:**
2	0h RO	**ABRT_10ADDR2_NOACK:**
1	0h RO	**ABRT_10ADDR1_NOACK:**
0	0h RO	**ABRT_7B_ADDR_NOACK:**

15.34 IC_SLV_DATA_NACK_ONLY — Offset 84h

Generate Slave Data NACK Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 84h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h NA	**RESERVED_1_31:** Reserved
0	0h RW	**NACK:**

15.35 IC_DMA_CR — Offset 88h

DMA Control Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 88h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h NA	**RESERVED_2_31:** Reserved
1	0h RW	**TDMAE:**
0	0h RW	**RDMAE:**

15.36 IC_DMA_TDLR — Offset 8Ch

DMA Transmit Data Level Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 8Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h NA	**RESERVED_8_31:** Reserved
7:0	0h RW	**DMATDL:**

15.37 IC_DMA_RDLR — Offset 90h

I2C Receive Data Level Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 90h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h NA	**RESERVED_8_31:** Reserved
7:0	0h RW	**DMARDL:**

15.38 IC_SDA_SETUP — Offset 94h

I2C SDA Setup Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 94h	64 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h NA	**RESERVED_8_31:** Reserved
7:0	64h RW	**SDA_SETUP:**

15.39 IC_ACK_GENERAL_CALL — Offset 98h

I2C ACK General Call Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 98h	1 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h NA	**RESERVED_1_31:** Reserved
0	1h RW	**ACK_GEN_CALL:**

15.40 IC_ENABLE_STATUS — Offset 9Ch

I2C Enable Status Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 9Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h NA	**RESERVED_3_31:** Reserved
2	0h RO	**SLV_RX_DATA_LOST:**
1	0h RO	**SLV_DISABLED_WHILE_BUSY:**
0	0h RO	**IC_EN:**

15.41 IC_FS_SPKLEN — Offset A0h

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + A0h	7 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h NA	**RESERVED_8_31:** Reserved
7:0	7h RW	**IC_FS_SPKLEN:**

15.42 IC_HS_SPKLEN — Offset A4h

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + A4h	2 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h NA	**RESERVED_8_31:** Reserved
7:0	2h RW	**IC_HS_SPKLEN:**

15.43 IC_CLR_RESTART_DET (IC_CLR_RESTRART_DET) — Offset A8h

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + A8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h NA	**RESERVED_1_31:** Reserved
0	0h RO	**IC_CLR_RESTART_DET:**

15.44 IC_COMP_PARAM_1 — Offset F4h

Component Parameter Register 1

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + F4h	3F3FEE h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h NA	**RESERVED_24_31:** Reserved
23:16	3Fh RO	**TX_BUFFER_DEPTH:**
15:8	3Fh RO	**RX_BUFFER_DEPTH:** The value of this register is derived from the IC_RX_BUFFER_DEPTH coreConsultant parameter.
7	1h RO	**ADD_ENCODED_PARAMS:**
6	1h RO	**HAS_DMA:**
5	1h RO	**INTR_IO:**
4	0h RO	**HC_COUNT_VALUES:**
3:2	3h RO	**MAX_SPEED_MODE:**
1:0	2h RO	**APB_DATA_WIDTH:**

15.45 IC_COMP_VERSION — Offset F8h

I2C Component Version Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + F8h	3132312A h

Bit Range	Default & Access	Field Name (ID): Description
31:0	3132312Ah RO	**IC_COMP_VERSION**

15.46 IC_COMP_TYPE — Offset FCh

I2C Component Type Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + FCh	44570140 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	44570140h RO	**IC_COMP_TYPE:**

15.47 reg_Reserved0 (RESERVED0) — Offset 200h

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 200h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**I2C_RESERVED0_REG:** i2c reserved0 register

15.48 reg_RESETS (RESETS) — Offset 204h

software reset

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 204h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h NA	**RESERVED:** Reserved
2	0h RW	**RESET_DMA:** DMA sw reset
1:0	0h RW	**RESET_IP:** i2c functional (serial) reset

15.49 reg_GENERAL (GENERAL) — Offset 208h

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 208h	5000000 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**SDA_MUX_SEL:** sda_mux_sel
30	0h RW	**SDA_SIGNAL_STATE:** sda_signal_state
29	0h RW	**SCL_MUX_SEL:** scl_mux_sel
28	0h RW	**SCL_SIGNAL_STATE:** scl_signal_state
27	0h RO	**I2C_SDA_RD_PRE_DRIVE:** i2c_sda_rd_pre_drive
26	1h RO	**I2C_SDA_RD_POST_DRIVE:** i2c_sda_rd_post_drive
25	0h RO	**I2C_SCL_RD_PRE_DRIVE:** i2c_scl_rd_pre_drive
24	1h RO	**I2C_SCL_RD_POST_DRIVE:** i2c_scl_rd_post_drive
23:5	0h NA	**reserved (I2C_GENERAL_REG_23_5):** reserved bits in i2c GEENRAL reg
4	0h RW	**TX_LASTBYTE_FLAG:** tx last byte (tx transaction completed) flag bit in i2c GENERAL reg
3	0h RW	**IO_VOLTAGE_SELECT:** io voltage select bit in i2c GENERAL reg
2:0	0h NA	**reserved1 (I2C_GENERAL_REG_0):** i2c GENERAL reg reserved bit 0

15.50 reg_ACTIVELTR (ACTIVELTR_VALUE) — Offset 210h

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 210h	800 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**NON_SNOOP_REQUIRMENT:** If the Requirement (bit 15) is clear, that indicates that the device has no LTR requirement for this type of traffic (i.e. it can wait for service indefinitely). If the 10-bit latency value is zero it indicates that the device cannot tolerate any delay and needs the best possible service/response time.
30:29	0h NA	**RESERVED_HIGH:** Reserved
28:26	0h RO	**NON_SNOOP_LATENCY_SCALE:** Support for codes 010 (1us) or 011 (32us) for Snoop Latency Scale (1us -> 32ms total span) only. Writes to this CSR which dont match those values will be dropped completely, next read will return previous value.
25:16	0h RO	**NON_SNOOP_VALUE:** 10-bit latency value
15	0h RW	**SNOOP_REQUIRMENT:** If the Requirement (bit 15) is clear, that indicates that the device has no LTR requirement for this type of traffic (i.e. it can wait for service indefinitely). If the 10-bit latency value is zero it indicates that the device cannot tolerate any delay and needs the best possible service/response time.
14:13	0h NA	**RESERVED_LOW:** Reserved
12:10	2h RW	**Snoop_latency_scale (I2C_SW_LTR_SNOOP_SCALE_REG_12_10):** Support for codes 010 (1us) or 011 (32us) for Snoop Latency Scale(1us -> 32ms total span) only. Writes to this CSR which dont match those values will be dropped completely, next read will return previous value.
9:0	0h RW	**SNOOP_VALUE:** 10-bit latency value

15.51 reg_IDLELTR (IDLELTR_VALUE) — Offset 214h

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 214h	800 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**NON_SNOOP_REQUIRMENT:** If the Requirement (bit 15) is clear, that indicates that the device has no LTR requirement for this type of traffic (i.e. it can wait for service indefinitely). If the 10-bit latency value is zero it indicates that the device cannot tolerate any delay and needs the best possible service/response time.
30:29	0h NA	**RESERVED_HIGH:** Reserved
28:26	0h RO	**NON_SNOOP_LATENCY_SCALE:** Support for codes 010 (1us) or 011 (32us) for Snoop Latency Scale (1us -> 32ms total span) only. Writes to this CSR which dont match those values will be dropped completely, next read will return previous value.
25:16	0h RO	**NON_SNOOP_VALUE:** 10-bit latency value

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**SNOOP_REQUIRMENT:** If the Requirement (bit 15) is clear, that indicates that the device has no LTR requirement for this type of traffic (i.e. it can wait for service indefinitely). If the 10-bit latency value is zero it indicates that the device cannot tolerate any delay and needs the best possible service/response time.
14:13	0h NA	**RESERVED_LOW:** Reserved
12:10	2h RW	**SNOOP_LATENCY_SCALE:** Support for codes 010 (1us) or 011 (32us) for Snoop Latency Scale(1us -> 32ms total span) only. Writes to this CSR which dont match those values will be dropped completely, next read will return previous value.
9:0	0h RW	**SNOOP_VALUE:** 10-bit latency value

15.52 reg_TX_ACK_COUNT (TX_ACK_COUNT) — Offset 218h

TX ack counts

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 218h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**tx_count_overflow (TX_ACK_COUNT_OVERFLOW):** count overflow indication
30:24	0h NA	**RESERVED:** Reserved
23:0	0h RO	**TX_ACK_COUNT:** Count ACK seen on Write commands

15.53 rx_ack_count (RX_BYTE_COUNT) — Offset 21Ch

Rx ACK counts

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 21Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**rx_byte_count_overflow (RX_ACK_COUNT_OVERFLOW):** count overflow indication
30:24	0h NA	**RESERVED:** Reserved
23:0	0h RO	**RX_ACK_COUNT:** Counts ACK seen on Read commands

15.54 TX_COMPLETE_INTR_STAT — Offset 220h

Interrupt status register for Tx Complete

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 220h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h NA	**RESERVED:** Reserved
1	0h RW	**tx_complete_intr_stat_mask (TX_INTR_STAT_MASK):** i2c tx completion interrupt mask
0	0h RO	**tx_complete_intr_stat (TX_INTR_STAT):** i2c tx completion interrupt status indication

15.55 TX_COMPLETE_INTR_CLR — Offset 224h

i2c tx complete interrupt clear indication

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 224h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h NA	**RESERVED:** Reserved
0	0h RO	**tx_completion_interrupt_clear (I2C_TX_COMPLETE_INTR_CLR_0):** i2c tx completion interrupt clear indication

15.56 reg_SW_SCRATCH_0 (SW_SCRATCH_0) — Offset 228h

SW scratch register 0

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 228h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SW_SCRATCH_0:** sw scratch reg 0

15.57 reg_SW_SCRATCH_1 (SW_SCRATCH_1) — Offset 22Ch

SW scratch register 1

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 22Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**i2c_sw_scratch_1_reg (SW_SCRATCH_1):** sw scratch reg 1

15.58 reg_SW_SCRATCH_2 (SW_SCRATCH_2) — Offset 230h

SW scratch register 2

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 230h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**i2c_sw_scratch_2_reg (SW_SCRATCH_2):** sw scratch reg 2

15.59 reg_SW_SCRATCH_3 (SW_SCRATCH_3) — Offset 234h

SW scratch register 3

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 234h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**i2c_sw_scratch_3_reg (SW_SCRATCH_3):** sw scratch reg 3

15.60 reg_CLOCK_GATE (CLOCK_GATE) — Offset 238h

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 238h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h NA	**RESERVED:** Reserved
3:2	0h RW	**SW_DMA_CLK_CTL:** dma clock gate cntrol bits, 00-hw clk en,01-rsv, 10-force off, 11-force on
1:0	0h RW	**SW_IP_CLK_CTL:** ip clock gate cntrol bits, 00-hw clk en,01-rsv, 10-force off, 11-force on

15.61 reg_Reserved1 (RESERVED1) — Offset 23Ch

i2c reserved1 register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 23Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**i2c_reserved1_reg (RESERVED1):** Reserved

15.62 reg_REMAP_ADDR_LO (REMAP_ADDR_LO) — Offset 240h

i2c remap address lo register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 240h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**I2C_REMAP_ADDR_LO_REG:** i2c remap addr lo reg

15.63 reg_REMAP_ADDR_HI (REMAP_ADDR_HI) — Offset 244h

i2c remap address hi register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 244h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**I2C_REMAP_ADDR_HI:** i2c remap addr hi reg

15.64 reg_Reserved2 (RESERVED2) — Offset 248h

i2c reserved2 register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 248h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**i2c_reserved2_reg (RESERVED2):** Reserved

15.65 reg_DEVIDLE_CONTROL (DEVIDLE_CONTROL) — Offset 24Ch

i2c dev idle control register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 24Ch	8 h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h NA	**RESERVED:** Reserved
4	0h RO	**INTR_REQ_CAPABLE:** intr_req_capable
3	1h RW/1C	**RESTORE_REQUIRED:** restore_required
2	0h RW	**DEVIDLE:** devidle
1	0h RO	**INTR_REQ:** intr_req
0	0h RO	**CMD_IN_PROGRESS:** cmd_in_progress

15.66 reg_CAPABLITIES (CAPABLITIES) — Offset 2FCh

i2c capapblities register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 2FCh	600 h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h NA	**RESERVED:** Reserved
10	1h NA	**VTOUCH_CAP:** vtouch_cap
9	1h RO	**SERIAL_CLK_FREQ:** 1-133 Mhz
8	0h RO	**IDMA_PRESENT:** 0= DMA present 1= DMA not present
7:4	0h RO	**INSTANCE_TYPE:** I2C, SPI or UART
3:0	0h RO	**INSTANCE_NUMBER:** i2c0 - 000 i2c1 - 001 .. i2c5- 101

15.67 SAR_LO0 — Offset 800h

The starting source address is programmed by software before the DMA channel is enabled, or by an LLI update before the start of the DMA transfer. While the DMA transfer is in progress, this register is updated to reflect the source address of the current transfer.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 800h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SAR_LO:** Current Source Address of DMA transfer. Updated after each source transfer. The SINC field in the CTL_LO[n]: Control Register determines whether the address increments or is left unchanged on every source transfer throughout the block transfer. When the channel is enabled (i.e. CH_EN is 1), the read back value will reflect the updated source transfer addresses. However, when the channel is disabled, the original programmed value will be reflected when reading this register. It's important to notice the following: 1. Once the block transfer is in progress (i.e. when channel is enabled), the read-back value correlates with the OCP Read Address that one would see in an OCP tracker. 2. If the read to this register comes during a block transfer, the LAST DMA Read address sent on the OCP before the register read is what's reflected in the read-back value. 3. If the last DMA read was a burst read (i.e. burst length > 1), the read-back value reflects the first address of the burst read since this is what gets sent on the OCP fabric. 4. If the read to the register occurred after the whole block got transferred, then the channel gets disabled and the returned value would be the original programmed value. 5. Since the read-back value is OCP based, only DW aligned addresses will be reflected (i.e. OCP Byte-Enable values would not be reflected) 6. Based on the above remarks, this value should be used as pseudo DMA read progress indicator when the channel is enabled and not an absolute one. Decrementing addresses are not supported

15.68 SAR_HI0 — Offset 804h

The starting source address is programmed by software before the DMA channel is enabled, or by an LLI update before the start of the DMA transfer. While the DMA transfer is in progress, this register is updated to reflect the source address of the current transfer.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 804h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SAR_HI:** Current Source Address of DMA transfer. Updated after each source transfer. The SINC field in the CTL_LO[n]: Control Register determines whether the address increments or is left unchanged on every source transfer throughout the block transfer. When the channel is enabled (i.e. CH_EN is 1), the read back value will reflect the updated source transfer addresses. However, when the channel is disabled, the original programmed value will be reflected when reading this register. It's important to notice the following: 1. Once the block transfer is in progress (i.e. when channel is enabled), the read-back value correlates with the OCP Read Address that one would see in an OCP tracker. 2. If the read to this register comes during a block transfer, the LAST DMA Read address sent on the OCP before the register read is what's reflected in the read-back value. 3. If the last DMA read was a burst read (i.e. burst length > 1), the read-back value reflects the first address of the burst read since this is what gets sent on the OCP fabric. 4. If the read to the register occurred after the whole block got transferred, then the channel gets disabled and the returned value would be the original programmed value. 5. Since the read-back value is OCP based, only DW aligned addresses will be reflected (i.e. OCP Byte-Enable values would not be reflected) 6. Based on the above remarks, this value should be used as pseudo DMA read progress indicator when the channel is enabled and not an absolute one. Decrementing addresses are not supported

15.69 DAR_LO0 — Offset 808h

The starting destination address is programmed by software before the DMA channel is enabled, or by an LLI update before the start of the DMA transfer. While the DMA transfer is in progress, this register is updated to reflect the destination address of the current transfer.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 808h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DAR_LO:** Current Destination Address of DMA transfer. Updated after each destination transfer. The DINC field in the CTL_LO[n]: Control Register determines whether the address increments or is left unchanged on every destination transfer throughout the block transfer. When the channel is enabled (i.e. CH_EN is 1), the read back value will reflect the updated destination transfer addresses. However, when the channel is disabled, the original programmed value will be reflected when reading this register. It's important to notice the following: 1. Once the block transfer is in progress (i.e. when channel is enabled), the read-back value correlates with the OCP Write Address that one would see in an OCP tracker. 2. If the read to this register comes during a block transfer, the LAST DMA Write address sent on the OCP before the register read is what's reflected in the read-back value. 3. If the last DMA write was a burst write (i.e. burst length > 1), the read-back value reflects the first address of the burst write since this is what gets sent on the OCP fabric. 4. If the read to the register occurred after the whole block got transferred, then the channel gets disabled and the returned value would be the original programmed value. 5. Since the read-back value is OCP based, only DW aligned addresses will be reflected (i.e. OCP Byte-Enable values would not be reflected) 6. Based on the above remarks, this value should be used as pseudo DMA write progress indicator when the channel is enabled and not an absolute one. Decrementing addresses are not supported

15.70 DAR_HI0 — Offset 80Ch

The starting destination address is programmed by software before the DMA channel is enabled, or by an LLI update before the start of the DMA transfer. While the DMA transfer is in progress, this register is updated to reflect the destination address of the current transfer.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 80Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DAR_HI:** Current Destination Address of DMA transfer. Updated after each destination transfer. The DINC field in the CTL_LO[n]: Control Register determines whether the address increments or is left unchanged on every destination transfer throughout the block transfer. When the channel is enabled (i.e. CH_EN is 1), the read back value will reflect the updated destination transfer addresses. However, when the channel is disabled, the original programmed value will be reflected when reading this register. It's important to notice the following: 1. Once the block transfer is in progress (i.e. when channel is enabled), the read-back value correlates with the OCP Write Address that one would see in an OCP tracker. 2. If the read to this register comes during a block transfer, the LAST DMA Write address sent on the OCP before the register read is what's reflected in the read-back value. 3. If the last DMA write was a burst write (i.e. burst length > 1), the read-back value reflects the first address of the burst write since this is what gets sent on the OCP fabric. 4. If the read to the register occurred after the whole block got transferred, then the channel gets disabled and the returned value would be the original programmed value. 5. Since the read-back value is OCP based, only DW aligned addresses will be reflected (i.e. OCP Byte-Enable values would not be reflected) 6. Based on the above remarks, this value should be used as pseudo DMA write progress indicator when the channel is enabled and not an absolute one. Decrementing addresses are not supported

15.71 LLP_LO0 — Offset 810h

Note: You need to program this register to point to the first Linked List Item (LLI) in memory prior to enabling the channel if block chaining is enabled The LLP register has two functions: The logical result of the equation LLP.LOC != 0 is used to set up the type of DMA transfer-single or multi-block. If LLP.LOC is set to 0x0, then transfers using

linked lists are not enabled. This register must be programmed prior to enabling the channel in order to set up the transfer type. LLP.LOC != 0 contains the pointer to the next LLI for block chaining using linked lists, The LLPx register can also point to the address where write-back of the control and source/destination status information occur after block completion.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 810h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**LOC:** Starting Address In Memory of next LLI if block chaining is enabled. Note that the two LSBs of the starting address are not stored because the address is assumed to be aligned to a 32-bit boundary. LLI accesses are always 32-bit accesses and cannot be changed or programmed to anything other than 32-bit.
1:0	0h NA	**RSVD2:** Reserved

15.72 LLP_HI0 — Offset 814h

Note: You need to program this register to point to the first Linked List Item (LLI) in memory prior to enabling the channel if block chaining is enabled The LLP register has two functions: The logical result of the equation LLP.LOC != 0 is used to set up the type of DMA transfer-single or multi-block. If LLP.LOC is set to 0x0, then transfers using linked lists are not enabled. This register must be programmed prior to enabling the channel in order to set up the transfer type. LLP.LOC != 0 contains the pointer to the next LLI for block chaining using linked lists, The LLPx register can also point to the address where write-back of the control and source/destination status information occur after block completion.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 814h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**LOC:** Starting Address In Memory of next LLI if block chaining is enabled. Note that the two LSBs of the starting address are not stored because the address is assumed to be aligned to a 32-bit boundary. LLI accesses are always 32-bit accesses and cannot be changed or programmed to anything other than 32-bit.
1:0	0h NA	**RSVD2:** Reserved

15.73 CTL_LO0 — Offset 818h

This register contains fields that control the DMA transfer.The CTL_LO register is part of the block descriptor (linked list item - LLI) when block chaining is enabled. It can be varied on a block-by-block basis within a DMA transfer when block chaining is enabled.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 818h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h NA	**RSVD:** Reserved
28	0h RW	**LLP_SRC_EN:** Block chaining is enabled on the source side only if the LLP_SRC_EN field is high and LLPn.LOC is non-zero and (LLP_EN == 1)
27	0h RW	**LLP_DST_EN:** Block chaining is enabled on the destination side only if the LLP_DST_EN field is high and LLPn.LOC is non-zero and (LLP_EN == 1)
26:22	0h RO	**RSVD1:** Reserved
21:20	0h RW	**TT_FC:** Transfer Type and Flow Control. The following transfer types are supported. Memory to Memory (00) Memory to Peripheral (01) Peripheral to Memory (10) Peripheral to Peripheral (11) Flow Control is always assigned to the DMA.
19	0h NA	**RSVD2:** Reserved
18	0h RW	**DST_SCATTER_EN:** Destination scatter enable bit: 0 = Scatter disabled 1 = Scatter enabled Scatter on the destination side is applicable only when the CTL_LOn.DINC bit indicates an incrementing address control.
17	0h RW	**SRC_GATHER_EN:** Source gather enable bit: 0 = Gather disabled 1 = Gather enabled Gather on the source side is applicable only when the CTL_LOn.SINC bit indicates an incrementing address control.
16:14	0h RW	**SRC_MSIZE:** Source Burst Transaction Length. Number of data items, each of width CTL_LOn.SRC_TR_WIDTH, to be read from the source.
13:11	0h RW	**DEST_MSIZE:** Destination Burst Transaction Length. Number of data items, each of width CTL_LOn.DST_TR_WIDTH, to be written to the destination.
10	0h RW	**SINC:** Source Address Increment. Indicates whether to increment or decrement the source address on every source transfer. If the device is fetching data from a source peripheral FIFO with a fixed address, then set this field to No change. 0 = Increment 1 = Fixed (No Change)
9	0h NA	**RSVD3:** Reserved
8	0h RW	**DINC:** Destination Address Increment. Indicates whether to increment or decrement the destination address on every destination transfer. If your device is writing data to a destination peripheral FIFO with a fixed address, then set this field to No change. 0 = Increment 1 = Fixed (No change)
7	0h NA	**RSVD4:** Reserved
6:4	0h RW	**SRC_TR_WIDTH:** Source Transfer Width. BURST_SIZE = (2 ^ MSIZE) 1. Transferred Bytes Per Burst = (BURST_SIZE * TR_WIDTH) 2. For incrementing addresses and (Transfer_Width < 4 Bytes), the MSIZE parameter is ignored since only single transactions are supported (due to OCP limitations). For a non-memory peripheral, this is typically the peripheral (source) FIFO width.
3:1	0h RW	**DST_TR_WIDTH:** Destination Transfer Width. BURST_SIZE = (2 ^ MSIZE) 1.Transferred Bytes Per Burst = (BURST_SIZE * TR_WIDTH) 2.For incrementing addresses and (Transfer_Width < 4 Bytes), the MSIZE parameter is ignored since only single transactions are supported (due to OCP limitations). For a non-memory peripheral, this is typically the peripheral (source) FIFO width.
0	0h RW	**INT_EN:** Interrupt Enable Bit. If set, then all interrupt-generating sources are enabled.

15.74 CTL_HI0 — Offset 81Ch

This register contains fields that control the DMA transfer. The CTL_HI register is part of the block descriptor (linked list item - LLI) when block chaining is enabled. It can be varied on a block-by-block basis within a DMA transfer when block chaining is enabled.If status write-back is enabled, CTL_HI is written to then control registers location of the LLI in system memory at the end of the block transfer.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 81Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RW	**CH_CLASS:** Channel Class. A Class of (N_CHNLS-1) is the highest priority, and 0 is the lowest. This field must be programmed within 0 to (N_CHNLS-1). A programmed value outside this range will cause erroneous behavior. ** See restriction in CH_WEIGHT below
28:18	0h RW	**CH_WEIGHT:** Channel Weight : Value of K assigns a weight of (K+1) in the round-robin arbitration between channels of the same class. A value of 0x7FF assigns an arbitration weight of 2048. Since K is from 0 to (2^11-1)=2047, Arbitration Weight ranges from 1 to 2048 **Restrictions : 1. CH_CLASS and CH_WEIGHT cannot be changed on the fly. Changes to either values for ANY channel require that ALL channels be quiescent in order to propagate those changes to the read and write arbiters without affecting their functionality. 2. Another possible way of achieving the quiescence requirement is to suspend ALL channels before changing the CH_CLASS and CH_WEIGHT. 3. Caution must be taken in descriptor-based (linked-list) multi-block transfers since the LLI.CTL_HI is one of the DW that needs to be read and loaded into the CTL_HI internal register. Hence, user needs to ensure that the CH_CLASS and CH_WEIGHT fields do not change from one descriptor to another nor do they disturb the aforementioned quiescence requirement.
17	0h RW	**DONE:** If status write-back is enabled, the upper word of the control register, CTL_HIn, is written to the control register location of the Linked List Item (LLI) in system memory at the end of the block transfer with the done bit set. Software can poll the LLI CTL_HI.DONE bit to see when a block transfer is complete. The LLI CTL_HI.DONE bit should be cleared when the linked lists are set up in memory prior to enabling the channel. The DMA will not transfer the block if the DONE bit in the LLI is not cleared.
16:0	0h RW	**BLOCK_TS:** Block Transfer Size (in Bytes). Since the DMA is always the flow controller, the user needs to write this field before the channel is enabled in order to indicate the block size. The number programmed into BLOCK_TS indicates the total number of bytes to transfer for every block transfer. Once the transfer starts (i.e. channel is enabled), the read-back value is the total number of bytes for which Read Commands have already been sent to the source. It doesnt mean Bytes that are already in the FIFO. However, when the channel is disabled, the original programmed value will be reflected when reading this register. Theoretical Byte Size range is from 0 to (2^17 -1) = (128 KB - 1).

15.75 SSTAT0 — Offset 820h

After each block transfer completes, hardware can retrieve the source status information from the address pointed to by the contents of the SSTATARx register. This status information is then stored in the SSTATx register and written out to the SSTATx register location of the LLI before the start of the next block. Note : This register is a temporary placeholder for the source status information on its way to the SSTATx register location of the LLI. The source status information should be retrieved by software from the SSTATx register location of the LLI, and not by a read of this register over the DMA slave interface.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 820h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SSTAT:** Source status information retrieved by hardware from the address pointed to by the contents of the Source Status Address Register. This register is a temporary placeholder for the source status information on its way to the SSTATn register location of the LLI. The source status information should be retrieved by software from the SSTATn register location of the LLI, and not by a read of this register over the DMA slave interface.

15.76 DSTAT0 — Offset 828h

After the completion of each block transfer, hardware can retrieve the destination status information from the address pointed to by the contents of the DSTATARx register. This status information is then stored in the DSTATx register and written out to the DSTATx register location of the LLI. Note : This register is a temporary placeholder for the destination status information on its way to the DSTATx register location of the LLI. The destination status information should be retrieved by software from the DSTATx register location of the LLI and not by a read of this register over the DMA slave interface.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 828h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DSTAT:** Destination status information retrieved by hardware from the address pointed to by the contents of the Destination Status Address Register. This register is a temporary placeholder for the destination status information on its way to the DSTATn register location of the LLI. The destination status information should be retrieved by software from the DSTATn register location of the LLI and not by a read of this register over the DMA slave interface.

15.77 SSTATAR_LO0 — Offset 830h

After the completion of each block transfer, hardware can retrieve the source status information from the address pointed to by the contents of the SSTATARx register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 830h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SSTATAR_LO:** Pointer from where hardware can fetch the source status information, which is registered in the SSTATn register and written out to the SSTATn register location of the LLI before the start of the next block.

15.78 SSTATAR_HI0 — Offset 834h

After the completion of each block transfer, hardware can retrieve the source status information from the address pointed to by the contents of the SSTATARx register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 834h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SSTATAR_HI:** Pointer from where hardware can fetch the source status information, which is registered in the SSTATn register and written out to the SSTATn register location of the LLI before the start of the next block.

15.79 DSTATAR_LO0 — Offset 838h

After the completion of each block transfer, hardware can retrieve the destination status information from the address pointed to by the contents of the DSTATARx register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 838h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DSTATAR_LO:** Pointer from where hardware can fetch the destination status information, which is registered in the DSTATn register and written out to the DSTATn register location of the LLI before the start of the next block.

15.80 DSTATAR_HI0 — Offset 83Ch

After the completion of each block transfer, hardware can retrieve the destination status information from the address pointed to by the contents of the DSTATARx register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 83Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DSTATAR_HI:** Pointer from where hardware can fetch the destination status information, which is registered in the DSTATn register and written out to the DSTATn register location of the LLI before the start of the next block.

15.81 CFG_LO0 — Offset 840h

This register contains fields that configure the DMA transfer. The channel configuration register remains fixed for all blocks of a multi-block transfer.This Register should be programmed to enabling the channel

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 840h	203 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**RELOAD_DST:** Automatic Destination Reload. The DARn register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated.
30	0h RW	**RELOAD_SRC:** Automatic Source Reload. The SARx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated.
29:22	0h NA	**RESERVED_29_22:** Reserved
21	0h RW	**SRC_OPT_BL:** Optimize Source Burst Length : 0 = Writes will use only BL=1 or BL=(2 ^ SRC_MSIZE) 1 = Writes will use (1 <= BL <= (2 ^ SRC_MSIZE)) *** This bit should be set to (0) if Source HW-Handshake is enabled
20	0h RW	**DST_OPT_BL:** Optimize Destination Burst Length : 0 = Writes will use only BL=1 or BL=(2 ^ DST_MSIZE) 1 = Writes will use (1 <= BL <= (2 ^ DST_MSIZE)) *** This bit should be set to (0) if Destination HW-Handshake is enabled
19	0h RW	**SRC_HS_POL:** Source Handshaking Interface Polarity. 0 = Active high 1 = Active low
18	0h RW	**DST_HS_POL:** Destination Handshaking Interface Polarity. 0 = Active high 1 = Active low
17:11	0h NA	**RESERVED_17_11:** Reserved
10	0h RW	**CH_DRAIN:** Forces channel FIFO to drain while in suspension. This bit has effect only when CH_SUSPEND ia asserted
9	1h RO	**FIFO_EMPTY:** Indicates if there is data left in the channel FIFO. Can be used in conjunction with CFGx.CH_SUSP to cleanly disable a channel. 1 = Channel FIFO empty 0 = Channel FIFO not empty
8	0h RW	**CH_SUSP:** Channel Suspend. Suspends all DMA data transfers from the source until this bit is cleared. There is no guarantee that the current transaction will complete. Can also be used in conjunction with CFGx.FIFO_EMPTY to cleanly disable a channel without losing any data. 0 = Not suspended. 1 = Suspend DMA transfer from the source.
7	0h RW	**SS_UPD_EN:** Source Status Update Enable. Source status information is fetched only from the location pointed to by the SSTATARn register, stored in the SSTATn register and written out to the SSTATn location of the LLI if SS_UPD_EN is high. This bit is ignored if (LLP_EN == 0) or (LLP_WB_EN == 0)

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**DS_UPD_EN:** Destination Status Update Enable. Destination status information is fetched only from the location pointed to by the DSTATARn register, stored in the DSTATn register and written out to the DSTATn location of the LLI if DS_UPD_EN is high. This bit is ignored if (LLP_EN == 0) or (LLP_WB_EN == 0)
5	0h RW	**CTL_HI_UPD_EN:** CTL_HI Update Enable. If set, the CTL_HI register is written out to the CTL_HIn location of the LLI. This bit is ignored if (LLP_EN == 0) or (LLP_WB_EN == 0)
4	0h NA	**RSVD_4_4:** Reserved
3	0h RW	**HSHAKE_NP_WR:** 0x1 : Issues Non-Posted writes on HW-Handshake on DMA Write Port 0x0 : Issues Posted writes on HW-Handshake on DMA Write Port (Except end-of-block writes which will be Non-Posted)
2	0h RW	**ALL_NP_WR:** 0x1 : Forces ALL writes to be Non-Posted on DMA Write Port 0x0 : Non-Posted Writes will only be used at end of block transfers and in HW-Handshake (if HW_NP_WR=1); for all other cases, Posted Writes will be used.
1	1h RW	**SRC_BURST_ALIGN:** 0x1 : SRC Burst Transfers are broken at a Burst Length aligned boundary 0x0 : SRC Burst Transfers are not broken at a Burst Length aligned boundary
0	1h RW	**DST_BURST_ALIGN:** 0x1 : DST Burst Transfers are broken at a Burst Length aligned boundary 0x0 : DST Burst Transfers are not broken at a Burst Length aligned boundary

15.82 CFG_HI0 — Offset 844h

This register contains fields that configure the DMA transfer. The channel configuration register remains fixed for all blocks of a multi-block transfer. This Register should be programmed to enabling the channel

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 844h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h NA	**RESERVED_31_28:** Reserved
27:18	0h RW	**WR_ISSUE_THD:** Write Issue Threshold. Used to relax the issue criterion for Writes. Value ranges from 0 to (2^10-1 = 1023) but should not exceed maximum Write burst size = (2 ^ DST_MSIZE)*TW.
17:8	0h RW	**RD_ISSUE_THD:** Read Issue Threshold. Used to relax the issue criterion for Reads. Value ranges from 0 to (2^10-1 = 1023) but should not exceed maximum Read burst size = (2 ^ SRC_MSIZE)*TW.
7:4	0h RW	**DST_PER:** Destination Peripheral ID : Assigns a hardware handshaking interface (0 - 15) to the destination of channel n. The channel can then communicate with the destination peripheral connected to that interface through the assigned hardware handshaking interface. NOTE: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.
3:0	0h RW	**SRC_PER:** Source Peripheral ID : Assigns a hardware handshaking interface (0 - 15) to the source of channel n. The channel can then communicate with the source peripheral connected to that interface through the assigned hardware handshaking interface. NOTE: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.

15.83 SGR0 — Offset 848h

The Source Gather register contains two fields: Source gather count field (SGRx.SGC). Specifies the number of contiguous source transfers of CTLx.SRC_TR_WIDTH between successive gather intervals. This is defined as a gather boundary. Source gather interval field (SGRx.SGI). Specifies the source address increment/decrement in multiples of CTLx.SRC_TR_WIDTH on a gather boundary when gather mode is enabled for the source transfer. The CTLx.SINC field controls whether the address increments or decrements. When the CTLx.SINC field indicates a fixed-address control, then the address remains constant throughout the transfer and the SGRx register is ignored.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 848h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW	**SGC:** Source gather count. Source contiguous transfer count between successive gather boundaries.
19:0	0h RW	**SGI:** Source gather interval.

15.84 DSR0 — Offset 850h

The Destination Scatter register contains two fields: Destination scatter count field (DSRx.DSC) . Specifies the number of contiguous destination transfers of CTLx.DST_TR_WIDTH between successive scatter boundaries. Destination scatter interval field (DSRx.DSI) . Specifies the destination address increment/decrement in multiples of CTLx.DST_TR_WIDTH on a scatter boundary when scatter mode is enabled for the destination transfer. The CTLx.DINC field controls whether the address increments or decrements. When the CTLx.DINC field indicates a fixed address control, then the address remains constant throughout the transfer and the DSRx register is ignored.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 850h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW	**DSC:** Destination scatter count. Destination contiguous transfer count between successive scatter boundaries.
19:0	0h RW	**DSI:** Destination scatter interval.

15.85 SAR_LO1 — Offset 858h

The starting source address is programmed by software before the DMA channel is enabled, or by an LLI update before the start of the DMA transfer. While the DMA transfer is in progress, this register is updated to reflect the source address of the current transfer.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 858h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SAR_LO:** Current Source Address of DMA transfer. Updated after each source transfer. The SINC field in the CTL_LO[n]: Control Register determines whether the address increments or is left unchanged on every source transfer throughout the block transfer. When the channel is enabled (i.e. CH_EN is 1), the read back value will reflect the updated source transfer addresses. However, when the channel is disabled, the original programmed value will be reflected when reading this register. It's important to notice the following: 1. Once the block transfer is in progress (i.e. when channel is enabled), the read-back value correlates with the OCP Read Address that one would see in an OCP tracker. 2. If the read to this register comes during a block transfer, the LAST DMA Read address sent on the OCP before the register read is what's reflected in the read-back value. 3. If the last DMA read was a burst read (i.e. burst length > 1), the read-back value reflects the first address of the burst read since this is what gets sent on the OCP fabric. 4. If the read to the register occurred after the whole block got transferred, then the channel gets disabled and the returned value would be the original programmed value. 5. Since the read-back value is OCP based, only DW aligned addresses will be reflected (i.e. OCP Byte-Enable values would not be reflected) 6. Based on the above remarks, this value should be used as pseudo DMA read progress indicator when the channel is enabled and not an absolute one. Decrementing addresses are not supported

15.86 SAR_HI1 — Offset 85Ch

The starting source address is programmed by software before the DMA channel is enabled, or by an LLI update before the start of the DMA transfer. While the DMA transfer is in progress, this register is updated to reflect the source address of the current transfer.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 85Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SAR_HI:** Current Source Address of DMA transfer. Updated after each source transfer. The SINC field in the CTL_LO[n]: Control Register determines whether the address increments or is left unchanged on every source transfer throughout the block transfer. When the channel is enabled (i.e. CH_EN is 1), the read back value will reflect the updated source transfer addresses. However, when the channel is disabled, the original programmed value will be reflected when reading this register. It's important to notice the following: 1. Once the block transfer is in progress (i.e. when channel is enabled), the read-back value correlates with the OCP Read Address that one would see in an OCP tracker. 2. If the read to this register comes during a block transfer, the LAST DMA Read address sent on the OCP before the register read is what's reflected in the read-back value. 3. If the last DMA read was a burst read (i.e. burst length > 1), the read-back value reflects the first address of the burst read since this is what gets sent on the OCP fabric. 4. If the read to the register occurred after the whole block got transferred, then the channel gets disabled and the returned value would be the original programmed value. 5. Since the read-back value is OCP based, only DW aligned addresses will be reflected (i.e. OCP Byte-Enable values would not be reflected) 6. Based on the above remarks, this value should be used as pseudo DMA read progress indicator when the channel is enabled and not an absolute one. Decrementing addresses are not supported

15.87 DAR_LO1 — Offset 860h

The starting destination address is programmed by software before the DMA channel is enabled, or by an LLI update before the start of the DMA transfer. While the DMA transfer is in progress, this register is updated to reflect the destination address of the current transfer.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 860h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DAR_LO:** Current Destination Address of DMA transfer. Updated after each destination transfer. The DINC field in the CTL_LO[n]: Control Register determines whether the address increments or is left unchanged on every destination transfer throughout the block transfer. When the channel is enabled (i.e. CH_EN is 1), the read back value will reflect the updated destination transfer addresses. However, when the channel is disabled, the original programmed value will be reflected when reading this register. It's important to notice the following: 1. Once the block transfer is in progress (i.e. when channel is enabled), the read-back value correlates with the OCP Write Address that one would see in an OCP tracker. 2. If the read to this register comes during a block transfer, the LAST DMA Write address sent on the OCP before the register read is what's reflected in the read-back value. 3. If the last DMA write was a burst write (i.e. burst length > 1), the read-back value reflects the first address of the burst write since this is what gets sent on the OCP fabric. 4. If the read to the register occurred after the whole block got transferred, then the channel gets disabled and the returned value would be the original programmed value. 5. Since the read-back value is OCP based, only DW aligned addresses will be reflected (i.e. OCP Byte-Enable values would not be reflected) 6. Based on the above remarks, this value should be used as pseudo DMA write progress indicator when the channel is enabled and not an absolute one. Decrementing addresses are not supported

15.88 DAR_HI1 — Offset 864h

The starting destination address is programmed by software before the DMA channel is enabled, or by an LLI update before the start of the DMA transfer. While the DMA transfer is in progress, this register is updated to reflect the destination address of the current transfer.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 864h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DAR_HI:** Current Destination Address of DMA transfer. Updated after each destination transfer. The DINC field in the CTL_LO[n]: Control Register determines whether the address increments or is left unchanged on every destination transfer throughout the block transfer. When the channel is enabled (i.e. CH_EN is 1), the read back value will reflect the updated destination transfer addresses. However, when the channel is disabled, the original programmed value will be reflected when reading this register. It's important to notice the following: 1. Once the block transfer is in progress (i.e. when channel is enabled), the read-back value correlates with the OCP Write Address that one would see in an OCP tracker. 2. If the read to this register comes during a block transfer, the LAST DMA Write address sent on the OCP before the register read is what's reflected in the read-back value. 3. If the last DMA write was a burst write (i.e. burst length > 1), the read-back value reflects the first address of the burst write since this is what gets sent on the OCP fabric. 4. If the read to the register occurred after the whole block got transferred, then the channel gets disabled and the returned value would be the original programmed value. 5. Since the read-back value is OCP based, only DW aligned addresses will be reflected (i.e. OCP Byte-Enable values would not be reflected) 6. Based on the above remarks, this value should be used as pseudo DMA write progress indicator when the channel is enabled and not an absolute one. Decrementing addresses are not supported

15.89 LLP_LO1 — Offset 868h

Note: You need to program this register to point to the first Linked List Item (LLI) in memory prior to enabling the channel if block chaining is enabled The LLP register has two functions: The logical result of the equation LLP.LOC != 0 is used to set up the type of DMA transfer-single or multi-block. If LLP.LOC is set to 0x0, then transfers using linked lists are not enabled. This register must be programmed prior to enabling the channel in order to set up the transfer type. LLP.LOC != 0 contains the pointer to the next LLI for block chaining using linked lists, The LLPx register can also point to the address where write-back of the control and source/destination status information occur after block completion.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 868h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**LOC:** Starting Address In Memory of next LLI if block chaining is enabled. Note that the two LSBs of the starting address are not stored because the address is assumed to be aligned to a 32-bit boundary. LLI accesses are always 32-bit accesses and cannot be changed or programmed to anything other than 32-bit.
1:0	0h NA	**RSVD2:** Reserved

15.90 LLP_HI1 — Offset 86Ch

Note: You need to program this register to point to the first Linked List Item (LLI) in memory prior to enabling the channel if block chaining is enabled The LLP register has two functions: The logical result of the equation LLP.LOC != 0 is used to set up the type of DMA transfer-single or multi-block. If LLP.LOC is set to 0x0, then transfers using linked lists are not enabled. This register must be programmed prior to enabling the channel in order to set up the transfer type. LLP.LOC != 0 contains the pointer to the next LLI for block chaining using linked lists, The LLPx register can also point to the address where write-back of the control and source/destination status information occur after block completion.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 86Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**LOC:** Starting Address In Memory of next LLI if block chaining is enabled. Note that the two LSBs of the starting address are not stored because the address is assumed to be aligned to a 32-bit boundary. LLI accesses are always 32-bit accesses and cannot be changed or programmed to anything other than 32-bit.
1:0	0h NA	**RSVD2:** Reserved

15.91 CTL_LO1 — Offset 870h

This register contains fields that control the DMA transfer.The CTL_LO register is part of the block descriptor (linked list item - LLI) when block chaining is enabled. It can be varied on a block-by-block basis within a DMA transfer when block chaining is enabled.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 870h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h NA	**RSVD:** Reserved
28	0h RW	**LLP_SRC_EN:** Block chaining is enabled on the source side only if the LLP_SRC_EN field is high and LLPn.LOC is non-zero and (LLP_EN == 1)
27	0h RW	**LLP_DST_EN:** Block chaining is enabled on the destination side only if the LLP_DST_EN field is high and LLPn.LOC is non-zero and (LLP_EN == 1)
26:22	0h RO	**RSVD1:** Reserved
21:20	0h RW	**TT_FC:** Transfer Type and Flow Control. The following transfer types are supported. Memory to Memory (00) Memory to Peripheral (01) Peripheral to Memory (10) Peripheral to Peripheral (11) Flow Control is always assigned to the DMA.
19	0h NA	**RSVD2:** Reserved

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW	**DST_SCATTER_EN:** Destination scatter enable bit: 0 = Scatter disabled 1 = Scatter enabled Scatter on the destination side is applicable only when the CTL_LOn.DINC bit indicates an incrementing address control.
17	0h RW	**SRC_GATHER_EN:** Source gather enable bit: 0 = Gather disabled 1 = Gather enabled Gather on the source side is applicable only when the CTL_LOn.SINC bit indicates an incrementing address control.
16:14	0h RW	**SRC_MSIZE:** Source Burst Transaction Length. Number of data items, each of width CTL_LOn.SRC_TR_WIDTH, to be read from the source.
13:11	0h RW	**DEST_MSIZE:** Destination Burst Transaction Length. Number of data items, each of width CTL_LOn.DST_TR_WIDTH, to be written to the destination.
10	0h RW	**SINC:** Source Address Increment. Indicates whether to increment or decrement the source address on every source transfer. If the device is fetching data from a source peripheral FIFO with a fixed address, then set this field to No change. 0 = Increment 1 = Fixed (No Change)
9	0h NA	**RSVD3:** Reserved
8	0h RW	**DINC:** Destination Address Increment. Indicates whether to increment or decrement the destination address on every destination transfer. If your device is writing data to a destination peripheral FIFO with a fixed address, then set this field to No change. 0 = Increment 1 = Fixed (No change)
7	0h NA	**RSVD4:** Reserved
6:4	0h RW	**SRC_TR_WIDTH:** Source Transfer Width. BURST_SIZE = (2 ^ MSIZE) 1. Transferred Bytes Per Burst = (BURST_SIZE * TR_WIDTH) 2. For incrementing addresses and (Transfer_Width < 4 Bytes), the MSIZE parameter is ignored since only single transactions are supported (due to OCP limitations). For a non-memory peripheral, this is typically the peripheral (source) FIFO width.
3:1	0h RW	**DST_TR_WIDTH:** Destination Transfer Width. BURST_SIZE = (2 ^ MSIZE) 1.Transferred Bytes Per Burst = (BURST_SIZE * TR_WIDTH) 2.For incrementing addresses and (Transfer_Width < 4 Bytes), the MSIZE parameter is ignored since only single transactions are supported (due to OCP limitations). For a non-memory peripheral, this is typically the peripheral (source) FIFO width.
0	0h RW	**INT_EN:** Interrupt Enable Bit. If set, then all interrupt-generating sources are enabled.

15.92 CTL_HI1 — Offset 874h

This register contains fields that control the DMA transfer. The CTL_HI register is part of the block descriptor (linked list item - LLI) when block chaining is enabled. It can be varied on a block-by-block basis within a DMA transfer when block chaining is enabled.If status write-back is enabled, CTL_HI is written to then control registers location of the LLI in system memory at the end of the block transfer.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 874h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RW	**CH_CLASS:** Channel Class. A Class of (N_CHNLS-1) is the highest priority, and 0 is the lowest. This field must be programmed within 0 to (N_CHNLS-1). A programmed value outside this range will cause erroneous behavior. ** See restriction in CH_WEIGHT below

Bit Range	Default & Access	Field Name (ID): Description
28:18	0h RW	**CH_WEIGHT:** Channel Weight : Value of K assigns a weight of (K+1) in the round-robin arbitration between channels of the same class. A value of 0x7FF assigns an arbitration weight of 2048. Since K is from 0 to (2^11-1)=2047, Arbitration Weight ranges from 1 to 2048 **Restrictions : 1. CH_CLASS and CH_WEIGHT cannot be changed on the fly. Changes to either values for ANY channel require that ALL channels be quiescent in order to propagate those changes to the read and write arbiters without affecting their functionality. 2. Another possible way of achieving the quiescence requirement is to suspend ALL channels before changing the CH_CLASS and CH_WEIGHT. 3. Caution must be taken in descriptor-based (linked-list) multi-block transfers since the LLI.CTL_HI is one of the DW that needs to be read and loaded into the CTL_HI internal register. Hence, user needs to ensure that the CH_CLASS and CH_WEIGHT fields do not change from one descriptor to another nor do they disturb the aforementioned quiescence requirement.
17	0h RW	**DONE:** If status write-back is enabled, the upper word of the control register, CTL_HIn, is written to the control register location of the Linked List Item (LLI) in system memory at the end of the block transfer with the done bit set. Software can poll the LLI CTL_HI.DONE bit to see when a block transfer is complete. The LLI CTL_HI.DONE bit should be cleared when the linked lists are set up in memory prior to enabling the channel. The DMA will not transfer the block if the DONE bit in the LLI is not cleared.
16:0	0h RW	**BLOCK_TS:** Block Transfer Size (in Bytes). Since the DMA is always the flow controller, the user needs to write this field before the channel is enabled in order to indicate the block size. The number programmed into BLOCK_TS indicates the total number of bytes to transfer for every block transfer. Once the transfer starts (i.e. channel is enabled), the read-back value is the total number of bytes for which Read Commands have already been sent to the source. It doesnt mean Bytes that are already in the FIFO. However, when the channel is disabled, the original programmed value will be reflected when reading this register. Theoretical Byte Size range is from 0 to (2^17 -1) = (128 KB - 1).

15.93 SSTAT1 — Offset 878h

After each block transfer completes, hardware can retrieve the source status information from the address pointed to by the contents of the SSTATARx register. This status information is then stored in the SSTATx register and written out to the SSTATx register location of the LLI before the start of the next block. Note : This register is a temporary placeholder for the source status information on its way to the SSTATx register location of the LLI. The source status information should be retrieved by software from the SSTATx register location of the LLI, and not by a read of this register over the DMA slave interface.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 878h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SSTAT:** Source status information retrieved by hardware from the address pointed to by the contents of the Source Status Address Register. This register is a temporary placeholder for the source status information on its way to the SSTATn register location of the LLI. The source status information should be retrieved by software from the SSTATn register location of the LLI, and not by a read of this register over the DMA slave interface.

15.94 DSTAT1 — Offset 880h

After the completion of each block transfer, hardware can retrieve the destination status information from the address pointed to by the contents of the DSTATARx register. This status information is then stored in the DSTATx register and written out to the DSTATx register location of the LLI. Note : This register is a temporary placeholder for the destination status information on its way to the DSTATx register location of the LLI. The destination status information should be retrieved by software from the DSTATx register location of the LLI and not by a read of this register over the DMA slave interface.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 880h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DSTAT:** Destination status information retrieved by hardware from the address pointed to by the contents of the Destination Status Address Register. This register is a temporary placeholder for the destination status information on its way to the DSTATn register location of the LLI. The destination status information should be retrieved by software from the DSTATn register location of the LLI and not by a read of this register over the DMA slave interface.

15.95 SSTATAR_LO1 — Offset 888h

After the completion of each block transfer, hardware can retrieve the source status information from the address pointed to by the contents of the SSTATARx register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 888h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SSTATAR_LO:** Pointer from where hardware can fetch the source status information, which is registered in the SSTATn register and written out to the SSTATn register location of the LLI before the start of the next block.

15.96 SSTATAR_HI1 — Offset 88Ch

After the completion of each block transfer, hardware can retrieve the source status information from the address pointed to by the contents of the SSTATARx register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 88Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SSTATAR_HI:** Pointer from where hardware can fetch the source status information, which is registered in the SSTATn register and written out to the SSTATn register location of the LLI before the start of the next block.

15.97 DSTATAR_LO1 — Offset 890h

After the completion of each block transfer, hardware can retrieve the destination status information from the address pointed to by the contents of the DSTATARx register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 890h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DSTATAR_LO:** Pointer from where hardware can fetch the destination status information, which is registered in the DSTATn register and written out to the DSTATn register location of the LLI before the start of the next block.

15.98 DSTATAR_HI1 — Offset 894h

After the completion of each block transfer, hardware can retrieve the destination status information from the address pointed to by the contents of the DSTATARx register.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 894h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DSTATAR_HI:** Pointer from where hardware can fetch the destination status information, which is registered in the DSTATn register and written out to the DSTATn register location of the LLI before the start of the next block.

15.99 CFG_LO1 — Offset 898h

This register contains fields that configure the DMA transfer. The channel configuration register remains fixed for all blocks of a multi-block transfer.This Register should be programmed to enabling the channel

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 898h	203 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**RELOAD_DST:** Automatic Destination Reload. The DARn register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated.
30	0h RW	**RELOAD_SRC:** Automatic Source Reload. The SARx register can be automatically reloaded from its initial value at the end of every block for multi-block transfers. A new block transfer is then initiated.
29:22	0h NA	**RESERVED_29_22:** Reserved
21	0h RW	**SRC_OPT_BL:** Optimize Source Burst Length : 0 = Writes will use only BL=1 or BL=(2 ^ SRC_MSIZE) 1 = Writes will use (1 <= BL <= (2 ^ SRC_MSIZE)) *** This bit should be set to (0) if Source HW-Handshake is enabled
20	0h RW	**DST_OPT_BL:** Optimize Destination Burst Length : 0 = Writes will use only BL=1 or BL=(2 ^ DST_MSIZE) 1 = Writes will use (1 <= BL <= (2 ^ DST_MSIZE)) *** This bit should be set to (0) if Destination HW-Handshake is enabled
19	0h RW	**SRC_HS_POL:** Source Handshaking Interface Polarity. 0 = Active high 1 = Active low
18	0h RW	**DST_HS_POL:** Destination Handshaking Interface Polarity. 0 = Active high 1 = Active low
17:11	0h NA	**RESERVED_17_11:** Reserved
10	0h RW	**CH_DRAIN:** Forces channel FIFO to drain while in suspension. This bit has effect only when CH_SUSPEND ia asserted
9	1h RO	**FIFO_EMPTY:** Indicates if there is data left in the channel FIFO. Can be used in conjunction with CFGx.CH_SUSP to cleanly disable a channel. 1 = Channel FIFO empty 0 = Channel FIFO not empty
8	0h RW	**CH_SUSP:** Channel Suspend. Suspends all DMA data transfers from the source until this bit is cleared. There is no guarantee that the current transaction will complete. Can also be used in conjunction with CFGx.FIFO_EMPTY to cleanly disable a channel without losing any data. 0 = Not suspended. 1 = Suspend DMA transfer from the source.
7	0h RW	**SS_UPD_EN:** Source Status Update Enable. Source status information is fetched only from the location pointed to by the SSTATARn register, stored in the SSTATn register and written out to the SSTATn location of the LLI if SS_UPD_EN is high. This bit is ignored if (LLP_EN == 0) or (LLP_WB_EN == 0)
6	0h RW	**DS_UPD_EN:** Destination Status Update Enable. Destination status information is fetched only from the location pointed to by the DSTATARn register, stored in the DSTATn register and written out to the DSTATn location of the LLI if DS_UPD_EN is high. This bit is ignored if (LLP_EN == 0) or (LLP_WB_EN == 0)
5	0h RW	**CTL_HI_UPD_EN:** CTL_HI Update Enable. If set, the CTL_HI register is written out to the CTL_HIn location of the LLI. This bit is ignored if (LLP_EN == 0) or (LLP_WB_EN == 0)
4	0h NA	**RSVD_4_4:** Reserved
3	0h RW	**HSHAKE_NP_WR:** 0x1 : Issues Non-Posted writes on HW-Handshake on DMA Write Port 0x0 : Issues Posted writes on HW-Handshake on DMA Write Port (Except end-of-block writes which will be Non-Posted)
2	0h RW	**ALL_NP_WR:** 0x1 : Forces ALL writes to be Non-Posted on DMA Write Port 0x0 : Non-Posted Writes will only be used at end of block transfers and in HW-Handshake (if HW_NP_WR=1); for all other cases, Posted Writes will be used.
1	1h RW	**SRC_BURST_ALIGN:** 0x1 : SRC Burst Transfers are broken at a Burst Length aligned boundary 0x0 : SRC Burst Transfers are not broken at a Burst Length aligned boundary
0	1h RW	**DST_BURST_ALIGN:** 0x1 : DST Burst Transfers are broken at a Burst Length aligned boundary 0x0 : DST Burst Transfers are not broken at a Burst Length aligned boundary

15.100 CFG_HI1 — Offset 89Ch

This register contains fields that configure the DMA transfer. The channel configuration register remains fixed for all blocks of a multi-block transfer. This Register should be programmed to enabling the channel

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 89Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h NA	**RESERVED_31_28:** Reserved
27:18	0h RW	**WR_ISSUE_THD:** Write Issue Threshold. Used to relax the issue criterion for Writes. Value ranges from 0 to (2^10-1 = 1023) but should not exceed maximum Write burst size = (2 ^ DST_MSIZE)*TW.
17:8	0h RW	**RD_ISSUE_THD:** Read Issue Threshold. Used to relax the issue criterion for Reads. Value ranges from 0 to (2^10-1 = 1023) but should not exceed maximum Read burst size = (2 ^ SRC_MSIZE)*TW.
7:4	0h RW	**DST_PER:** Destination Peripheral ID : Assigns a hardware handshaking interface (0 - 15) to the destination of channel n. The channel can then communicate with the destination peripheral connected to that interface through the assigned hardware handshaking interface. NOTE: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.
3:0	0h RW	**SRC_PER:** Source Peripheral ID : Assigns a hardware handshaking interface (0 - 15) to the source of channel n. The channel can then communicate with the source peripheral connected to that interface through the assigned hardware handshaking interface. NOTE: For correct DMA operation, only one peripheral (source or destination) should be assigned to the same handshaking interface.

15.101 SGR1 — Offset 8A0h

The Source Gather register contains two fields: Source gather count field (SGRx.SGC). Specifies the number of contiguous source transfers of CTLx.SRC_TR_WIDTH between successive gather intervals. This is defined as a gather boundary. Source gather interval field (SGRx.SGI). Specifies the source address increment/decrement in multiples of CTLx.SRC_TR_WIDTH on a gather boundary when gather mode is enabled for the source transfer. The CTLx.SINC field controls whether the address increments or decrements. When the CTLx.SINC field indicates a fixed-address control, then the address remains constant throughout the transfer and the SGRx register is ignored.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 8A0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW	**SGC:** Source gather count. Source contiguous transfer count between successive gather boundaries.
19:0	0h RW	**SGI:** Source gather interval.

15.102 DSR1 — Offset 8A8h

The Destination Scatter register contains two fields: Destination scatter count field (DSRx.DSC) . Specifies the number of contiguous destination transfers of CTLx.DST_TR_WIDTH between successive scatter boundaries. Destination scatter interval field (DSRx.DSI) . Specifies the destination address increment/decrement in multiples of CTLx.DST_TR_WIDTH on a scatter boundary when scatter mode is enabled for the destination transfer. The CTLx.DINC field controls whether the address increments or decrements. When the CTLx.DINC field indicates a fixed address control, then the address remains constant throughout the transfer and the DSRx register is ignored.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + 8A8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW	**DSC:** Destination scatter count. Destination contiguous transfer count between successive scatter boundaries.
19:0	0h RW	**DSI:** Destination scatter interval.

15.103 RAWTFR — Offset AC0h

Interrupt events are stored in these Raw Interrupt Status registers before masking: RawBlock, RawDstTran, RawErr, RawSrcTran, and RawTfr. Each Raw Interrupt Status register has a bit allocated per channel, for example, RawTfr(2) is the Channel 2 raw transfer complete interrupt. Each bit in these registers is cleared by writing a 1 to the corresponding location in the ClearTfr, ClearBlock, ClearSrcTran, ClearDstTran, ClearErr registers The following RAW registers are available in the DMA RawTfr - Raw Status for Transfer Interrupts RawBlock - Raw Status for Block Interrupts Register RawSrcTran - Raw Status for Source Transaction Interrupts Register RawDstTran - Raw Status for Destination Transaction Interrupts Register RawErr - Raw Status for Error Interrupts Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + AC0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h NA	**RSVD:** Reserved
1:0	0h RO	**RAW:** Raw interrupt status

15.104 RAWBLOCK — Offset AC8h

Interrupt events are stored in these Raw Interrupt Status registers before masking: RawBlock, RawDstTran, RawErr, RawSrcTran, and RawTfr. Each Raw Interrupt Status register has a bit allocated per channel, for example, RawTfr(2) is the Channel 2 raw transfer complete interrupt. Each bit in these registers is cleared by writing a 1 to the corresponding location in the ClearTfr, ClearBlock, ClearSrcTran, ClearDstTran, ClearErr registers The following RAW registers are available in the DMA RawTfr - Raw Status for Transfer Interrupts RawBlock - Raw Status for Block Interrupts Register RawSrcTran - Raw Status for Source Transaction Interrupts Register RawDstTran - Raw Status for Destination Transaction Interrupts Register RawErr - Raw Status for Error Interrupts Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + AC8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h NA	**RSVD:** Reserved
1:0	0h RO	**RAW:** Raw interrupt status

15.105 RAWSRCTRAN — Offset AD0h

Interrupt events are stored in these Raw Interrupt Status registers before masking: RawBlock, RawDstTran, RawErr, RawSrcTran, and RawTfr. Each Raw Interrupt Status register has a bit allocated per channel, for example, RawTfr(2) is the Channel 2 raw transfer complete interrupt. Each bit in these registers is cleared by writing a 1 to the corresponding location in the ClearTfr, ClearBlock, ClearSrcTran, ClearDstTran, ClearErr registers The following RAW registers are available in the DMA RawTfr - Raw Status for Transfer Interrupts RawBlock - Raw Status for Block Interrupts Register RawSrcTran - Raw Status for Source Transaction Interrupts Register RawDstTran - Raw Status for Destination Transaction Interrupts Register RawErr - Raw Status for Error Interrupts Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + AD0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h NA	**RSVD:** Reserved
1:0	0h RO	**RAW:** Raw interrupt status

15.106 RAWDSTTRAN — Offset AD8h

Interrupt events are stored in these Raw Interrupt Status registers before masking: RawBlock, RawDstTran, RawErr, RawSrcTran, and RawTfr. Each Raw Interrupt Status register has a bit allocated per channel, for example, RawTfr(2) is the Channel 2 raw transfer complete interrupt. Each bit in these registers is cleared by writing a 1 to the corresponding location in the ClearTfr, ClearBlock, ClearSrcTran, ClearDstTran, ClearErr registers The following RAW registers are available in the DMA RawTfr - Raw Status for Transfer Interrupts RawBlock - Raw Status for Block Interrupts Register RawSrcTran - Raw Status for Source Transaction Interrupts Register RawDstTran - Raw Status for Destination Transaction Interrupts Register RawErr - Raw Status for Error Interrupts Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + AD8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h NA	**RSVD:** Reserved
1:0	0h RO	**RAW:** Raw interrupt status

15.107 RAWERR — Offset AE0h

Interrupt events are stored in these Raw Interrupt Status registers before masking: RawBlock, RawDstTran, RawErr, RawSrcTran, and RawTfr. Each Raw Interrupt Status register has a bit allocated per channel, for example, RawTfr(2) is the Channel 2 raw transfer complete interrupt. Each bit in these registers is cleared by writing a 1 to the corresponding location in the ClearTfr, ClearBlock, ClearSrcTran, ClearDstTran, ClearErr registers The following RAW registers are available in the DMA RawTfr - Raw Status for Transfer Interrupts RawBlock - Raw Status for Block Interrupts Register RawSrcTran - Raw Status for Source Transaction Interrupts Register RawDstTran - Raw Status for Destination Transaction Interrupts Register RawErr - Raw Status for Error Interrupts Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + AE0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h NA	**RSVD:** Reserved
1:0	0h RO	**RAW:** Raw interrupt status

15.108 STATUSTFR — Offset AE8h

All interrupt events from all channels are stored in these Interrupt Status registers after masking: statusBlock, StatusDstTran, StatusErr, StatusSrcTran, and StatusTfr. Each Interrupt Status register has a bit allocated per channel, for example, StatusTfr(2) is the Channel 2 status transfer complete interrupt. The contents of these registers are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMA.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + AE8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h NA	**RSVD:** Reserved
1:0	0h RO	**STATUS:** Interrupt status

15.109 STATUSBLOCK — Offset AF0h

All interrupt events from all channels are stored in these Interrupt Status registers after masking: statusBlock, StatusDstTran, StatusErr, StatusSrcTran, and StatusTfr. Each Interrupt Status register has a bit allocated per channel, for example, StatusTfr(2) is the Channel 2 status transfer complete interrupt. The contents of these registers are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMA.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + AF0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h NA	**RSVD:** Reserved
1:0	0h RO	**STATUS:** Interrupt status

15.110 STATUSSRCTRAN — Offset AF8h

All interrupt events from all channels are stored in these Interrupt Status registers after masking: statusBlock, StatusDstTran, StatusErr, StatusSrcTran, and StatusTfr. Each Interrupt Status register has a bit allocated per channel, for example, StatusTfr(2) is the Channel 2 status transfer complete interrupt. The contents of these registers are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMA.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + AF8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h NA	**RSVD:** Reserved
1:0	0h RO	**STATUS:** Interrupt status

15.111 STATUSDSTTRAN — Offset B00h

All interrupt events from all channels are stored in these Interrupt Status registers after masking: statusBlock, StatusDstTran, StatusErr, StatusSrcTran, and StatusTfr. Each Interrupt Status register has a bit allocated per channel, for example, StatusTfr(2) is the Channel 2 status transfer complete interrupt. The contents of these registers are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMA.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + B00h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h NA	**RSVD:** Reserved
1:0	0h RO	**STATUS:** Interrupt status

15.112 STATUSERR — Offset B08h

All interrupt events from all channels are stored in these Interrupt Status registers after masking: statusBlock, StatusDstTran, StatusErr, StatusSrcTran, and StatusTfr. Each Interrupt Status register has a bit allocated per channel, for example, StatusTfr(2) is the Channel 2 status transfer complete interrupt. The contents of these registers are used to generate the interrupt signals (int or int_n bus, depending on interrupt polarity) leaving the DMA.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + B08h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h NA	**RSVD:** Reserved
1:0	0h RO	**STATUS:** Interrupt status

15.113 MASKTFR — Offset B10h

The contents of the Raw Status registers are masked with the contents of the Mask registers: MaskBlock, MaskDstTran, MaskErr, MaskSrcTran, and MaskTfr. Each Interrupt Mask register has a bit allocated per channel, for example, MaskTfr(2) is the mask bit for the Channel 2 transfer complete interrupt. When the source peripheral of DMA channel i is memory, then the source transaction complete interrupt, MaskSrcTran(i), must be masked to prevent an erroneous triggering of an interrupt on the int_combined signal. Similarly, when the destination peripheral of DMA channel i is memory, then the destination transaction complete interrupt, MaskDstTran(i), must be masked to prevent an erroneous triggering of an interrupt on the int_combined(_n) signal. A channel INT_MASK bit will be written only if the corresponding mask write enable bit in the INT_MASK_WE field is asserted on the same OCP write transfer. This allows software to set a mask bit without performing a read-modified write operation. For example, writing hex 01x1 to the MaskTfr register writes a 1 into MaskTfr(0), while MaskTfr(7:1) remains unchanged. Writing hex 00xx leaves MaskTfr(7:0) unchanged. Writing a 1 to any bit in these registers unmasks the corresponding interrupt, thus allowing the DMA to set the appropriate bit in the Status registers and int_* port signals.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + B10h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h NA	**RSVD2:** Reserved
15:10	0h NA	**RSVD1:** Reserved
9:8	0h WO	**INT_MASK_WE:** Interrupt Mask Write Enable 0 = write disabled 1 = write enabled
7:2	0h NA	**RSVD0:** Reserved
1:0	0h RW	**INT_MASK:** Interrupt mask 0-mask 1-unmask

15.114 MASKBLOCK — Offset B18h

The contents of the Raw Status registers are masked with the contents of the Mask registers: MaskBlock, MaskDstTran, MaskErr, MaskSrcTran, and MaskTfr. Each Interrupt Mask register has a bit allocated per channel, for example, MaskTfr(2) is the mask bit for the Channel 2 transfer complete interrupt. When the source peripheral of DMA

channel i is memory, then the source transaction complete interrupt, MaskSrcTran(i), must be masked to prevent an erroneous triggering of an interrupt on the int_combined signal. Similarly, when the destination peripheral of DMA channel i is memory, then the destination transaction complete interrupt, MaskDstTran(i), must be masked to prevent an erroneous triggering of an interrupt on the int_combined(_n) signal. A channel INT_MASK bit will be written only if the corresponding mask write enable bit in the INT_MASK_WE field is asserted on the same OCP write transfer. This allows software to set a mask bit without performing a read-modified write operation. For example, writing hex 01x1 to the MaskTfr register writes a 1 into MaskTfr(0), while MaskTfr(7:1) remains unchanged. Writing hex 00xx leaves MaskTfr(7:0) unchanged. Writing a 1 to any bit in these registers unmasks the corresponding interrupt, thus allowing the DMA to set the appropriate bit in the Status registers and int_* port signals.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + B18h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h NA	**RSVD2:** Reserved
15:10	0h NA	**RSVD1:** Reserved
9:8	0h WO	**INT_MASK_WE:** Interrupt Mask Write Enable 0 = write disabled 1 = write enabled
7:2	0h NA	**RSVD0:** Reserved
1:0	0h RW	**INT_MASK:** Interrupt mask 0-mask 1-unmask

15.115 MASKSRCTRAN — Offset B20h

The contents of the Raw Status registers are masked with the contents of the Mask registers: MaskBlock, MaskDstTran, MaskErr, MaskSrcTran, and MaskTfr. Each Interrupt Mask register has a bit allocated per channel, for example, MaskTfr(2) is the mask bit for the Channel 2 transfer complete interrupt. When the source peripheral of DMA channel i is memory, then the source transaction complete interrupt, MaskSrcTran(i), must be masked to prevent an erroneous triggering of an interrupt on the int_combined signal. Similarly, when the destination peripheral of DMA channel i is memory, then the destination transaction complete interrupt, MaskDstTran(i), must be masked to prevent an erroneous triggering of an interrupt on the int_combined(_n) signal. A channel INT_MASK bit will be written only if the corresponding mask write enable bit in the INT_MASK_WE field is asserted on the same OCP write transfer. This allows software to set a mask bit without performing a read-modified write operation. For example, writing hex 01x1 to the MaskTfr register writes a 1 into MaskTfr(0), while MaskTfr(7:1) remains unchanged. Writing hex 00xx leaves MaskTfr(7:0) unchanged. Writing a 1 to any bit in these registers unmasks the corresponding interrupt, thus allowing the DMA to set the appropriate bit in the Status registers and int_* port signals.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + B20h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h NA	**RSVD2:** Reserved
15:10	0h NA	**RSVD1:** Reserved
9:8	0h WO	**INT_MASK_WE:** Interrupt Mask Write Enable 0 = write disabled 1 = write enabled
7:2	0h NA	**RSVD0:** Reserved
1:0	0h RW	**INT_MASK:** Interrupt mask 0-mask 1-unmask

15.116 MASKDSTTRAN — Offset B28h

The contents of the Raw Status registers are masked with the contents of the Mask registers: MaskBlock, MaskDstTran, MaskErr, MaskSrcTran, and MaskTfr. Each Interrupt Mask register has a bit allocated per channel, for example, MaskTfr(2) is the mask bit for the Channel 2 transfer complete interrupt. When the source peripheral of DMA channel i is memory, then the source transaction complete interrupt, MaskSrcTran(i), must be masked to prevent an erroneous triggering of an interrupt on the int_combined signal. Similarly, when the destination peripheral of DMA channel i is memory, then the destination transaction complete interrupt, MaskDstTran(i), must be masked to prevent an erroneous triggering of an interrupt on the int_combined(_n) signal. A channel INT_MASK bit will be written only if the corresponding mask write enable bit in the INT_MASK_WE field is asserted on the same OCP write transfer. This allows software to set a mask bit without performing a read-modified write operation. For example, writing hex 01x1 to the MaskTfr register writes a 1 into MaskTfr(0), while MaskTfr(7:1) remains unchanged. Writing hex 00xx leaves MaskTfr(7:0) unchanged. Writing a 1 to any bit in these registers unmasks the corresponding interrupt, thus allowing the DMA to set the appropriate bit in the Status registers and int_* port signals.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + B28h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h NA	**RSVD2:** Reserved
15:10	0h NA	**RSVD1:** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h WO	**INT_MASK_WE:** Interrupt Mask Write Enable 0 = write disabled 1 = write enabled
7:2	0h NA	**RSVD0:** Reserved
1:0	0h RW	**INT_MASK:** Interrupt mask 0-mask 1-unmask

15.117 MASKERR — Offset B30h

The contents of the Raw Status registers are masked with the contents of the Mask registers: MaskBlock, MaskDstTran, MaskErr, MaskSrcTran, and MaskTfr. Each Interrupt Mask register has a bit allocated per channel, for example, MaskTfr(2) is the mask bit for the Channel 2 transfer complete interrupt. When the source peripheral of DMA channel i is memory, then the source transaction complete interrupt, MaskSrcTran(i), must be masked to prevent an erroneous triggering of an interrupt on the int_combined signal. Similarly, when the destination peripheral of DMA channel i is memory, then the destination transaction complete interrupt, MaskDstTran(i), must be masked to prevent an erroneous triggering of an interrupt on the int_combined(_n) signal. A channel INT_MASK bit will be written only if the corresponding mask write enable bit in the INT_MASK_WE field is asserted on the same OCP write transfer. This allows software to set a mask bit without performing a read-modified write operation. For example, writing hex 01x1 to the MaskTfr register writes a 1 into MaskTfr(0), while MaskTfr(7:1) remains unchanged. Writing hex 00xx leaves MaskTfr(7:0) unchanged. Writing a 1 to any bit in these registers unmasks the corresponding interrupt, thus allowing the DMA to set the appropriate bit in the Status registers and int_* port signals.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + B30h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h NA	**RSVD2:** Reserved
15:10	0h NA	**RSVD1:** Reserved
9:8	0h WO	**INT_MASK_WE:** Interrupt Mask Write Enable 0 = write disabled 1 = write enabled
7:2	0h NA	**RSVD0:** Reserved
1:0	0h RW	**INT_MASK:** Interrupt mask 0-mask 1-unmask

15.118 CLEARTFR — Offset B38h

Each bit in the Raw Status and Status registers is cleared on the same cycle by writing a 1 to the corresponding location in the Clear registers: ClearBlock, ClearDstTran, ClearErr, ClearSrcTran, and ClearTfr. Each Interrupt Clear register has a bit allocated per channel, for example, ClearTfr(2) is the clear bit for the Channel 2 transfer complete interrupt. Writing a 0 has no effect. These registers are not readable.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + B38h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h NA	**RSVD:** Reserved
1:0	0h WO	**CLEAR:** Interrupt clear. 0 = no effect 1 = clear interrupt

15.119 CLEARBLOCK — Offset B40h

Each bit in the Raw Status and Status registers is cleared on the same cycle by writing a 1 to the corresponding location in the Clear registers: ClearBlock, ClearDstTran, ClearErr, ClearSrcTran, and ClearTfr. Each Interrupt Clear register has a bit allocated per channel, for example, ClearTfr(2) is the clear bit for the Channel 2 transfer complete interrupt. Writing a 0 has no effect. These registers are not readable.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + B40h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h NA	**RSVD:** Reserved
1:0	0h WO	**CLEAR:** Interrupt clear. 0 = no effect 1 = clear interrupt

15.120 CLEARSRCTRAN — Offset B48h

Each bit in the Raw Status and Status registers is cleared on the same cycle by writing a 1 to the corresponding location in the Clear registers: ClearBlock, ClearDstTran, ClearErr, ClearSrcTran, and ClearTfr. Each Interrupt Clear register has a bit allocated per channel, for example, ClearTfr(2) is the clear bit for the Channel 2 transfer complete interrupt. Writing a 0 has no effect. These registers are not readable.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + B48h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h NA	**RSVD:** Reserved
1:0	0h WO	**CLEAR:** Interrupt clear. 0 = no effect 1 = clear interrupt

15.121 CLEARDSTTRAN — Offset B50h

Each bit in the Raw Status and Status registers is cleared on the same cycle by writing a 1 to the corresponding location in the Clear registers: ClearBlock, ClearDstTran, ClearErr, ClearSrcTran, and ClearTfr. Each Interrupt Clear register has a bit allocated per channel, for example, ClearTfr(2) is the clear bit for the Channel 2 transfer complete interrupt. Writing a 0 has no effect. These registers are not readable.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + B50h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h NA	**RSVD:** Reserved
1:0	0h WO	**CLEAR:** Interrupt clear. 0 = no effect 1 = clear interrupt

15.122 CLEARERR — Offset B58h

Each bit in the Raw Status and Status registers is cleared on the same cycle by writing a 1 to the corresponding location in the Clear registers: ClearBlock, ClearDstTran, ClearErr, ClearSrcTran, and ClearTfr. Each Interrupt Clear register has a bit allocated per channel, for example, ClearTfr(2) is the clear bit for the Channel 2 transfer complete interrupt. Writing a 0 has no effect. These registers are not readable.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + B58h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h NA	**RSVD:** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h WO	**CLEAR:** Interrupt clear. 0 = no effect 1 = clear interrupt

15.123 STATUSINT — Offset B60h

The contents of each of the five Status registers StatusTfr, StatusBlock, StatusSrcTran, StatusDstTran,StatusErr is ORed to produce a single bit for each interrupt type in the Combined Status register (StatusInt). This register is read-only.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + B60h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h NA	**RSVD:** Reserved
4	0h RO	**ERR:** OR of the contents of StatusErr register.
3	0h RO	**DSTT:** OR of the contents of StatusDst register.
2	0h RO	**SRCT:** OR of the contents of StatusSrcTran register
1	0h RO	**BLOCK:** OR of the contents of StatusBlock register.
0	0h RO	**TFR:** OR of the contents of StatusTfr register.

15.124 DMACFGREG — Offset B98h

This register is used to enable the DMA, which must be done before any channel activity can begin. If the global channel enable bit is cleared while any channel is still active, then DmaCfgReg.DMA_EN still returns 1 to indicate that there are channels still active until hardware has terminated all activity on all channels, at which point the DmaCfgReg.DMA_EN bit returns 0.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + B98h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h NA	**RSVD:** Reserved
0	0h RW	**DMA_EN:** DMA Enable bit. 0 = DMA Disabled 1 = DMA Enabled

15.125 CHENREG — Offset BA0h

This is the DMA Channel Enable Register. If software needs to set up a new channel, then it can read this register in order to find out which channels are currently inactive, it can then enable an inactive channel with the required priority. All bits of this register are cleared to 0 when the global DMA channel enable bit, DmaCfgReg(0), is 0. When the global channel enable bit is 0, then a write to the ChEnReg register is ignored and a read will always read back 0. The channel enable bit, ChEnReg.CH_EN, is written only if the corresponding channel write enable bit, ChEnReg.CH_EN_WE, is asserted on the same OCP write transfer. For example, writing hex 01x1 writes a 1 into ChEnReg(0), while ChEnReg(7:1) remains unchanged. Writing hex 00xx leaves ChEnReg(7:0) unchanged. Note that a read-modified write is not required.

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + BA0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h NA	**RSVD2:** Reserved
15:10	0h NA	**RSVD1:** Reserved
9:8	0h WO	**CH_EN_WE:** Channel enable write enable.
7:2	0h NA	**RSVD0:** Reserved
1:0	0h RW	**CH_EN:** Enables/Disables the channel. Setting this bit enables a channel, clearing this bit disables the channel. 0 = Disable the Channel 1 = Enable the Channel The ChEnReg.CH_EN bit is automatically cleared by hardware to disable the channel after the last OCP transfer of the DMA transfer to the destination has completed. Software can therefore poll this bit to determine when this channel is free for a new DMA transfer.

15.126 RESERVED0_CPL — Offset BB8h

reserved

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + BB8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**RESERVED_31_0:** Reserved

15.127 RESERVED0_CPH — Offset BBCh

reserved

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + BBCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**RESERVED_31_0:** Reserved

15.128 RESERVED1_CPL — Offset BC0h

reserved

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + BC0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**RESERVED_31_0:** Reserved

15.129 RESERVED1_CPH — Offset BC4h

reserved

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + BC4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**RESERVED_31_0:** Reserved

15.130 RESERVED0_FPL — Offset C00h

reserved

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + C00h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**RESERVED_31_0:** Reserved

15.131 RESERVED0_FPH — Offset C04h

reserved

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + C04h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**RESERVED_31_0:** Reserved

15.132 RESERVED1_FPL — Offset C08h

reserved

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + C08h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**RESERVED_31_0:** Reserved

15.133 RESERVED1_FPH — Offset C0Ch

reserved

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + C0Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**RESERVED_31_0:** Reserved

15.134 SAI_ERR — Offset C10h

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + C10h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h NA	**RSVD:** Reserved
1:0	0h RO	**CH_SAI_ERR:** 1: SAI Error occurred on Ch [n] 0: No SAI Error occurred on Ch [n] SAI_ERROR[n] is set by HW on SAI violation for channel [n]. All bits get cleared by SW when reading this register

15.135 GLOBAL_CFG — Offset C18h

GLOBAL_CFG: GLOBAL DMA Configuration Register

Type	Size	Offset	Default
MEM	32 bit	[B:0, D:22, F:0] LPIOBAR + C18h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RW	**SCRATCH_PAD:** This field will have all bits with R/W attribute to be used for future configuration and de-feature bits.
0	0h RW	**ERR_ILL_REG:** 0x1 : Issue ERR response on reading illegal (non-existing) registers 0x0 : Issue DVA response on reading illegal (non-existing) registers

§ §

16 USB-Host (xHCI) Registers

This chapter documents the registers in Bus: 0, Device 21, Function 0.

16.1 Vendor ID (VID) — Offset 0h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:21, F:0] + 0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RO	**Vendor ID (VID):**

16.2 Device ID (DID) — Offset 2h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:21, F:0] + 2h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RO/V	**Device ID (DID)**

16.3 Command (CMD) — Offset 4h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:21, F:0] + 4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:11	0h RO	**Reserved (RSVD):** Reserved
10	0h RW	**Interrupt Disable (ID):** When cleared to 0, the function is capable of generating interrupts. When 1, the function can not generate its interrupt to the interrupt controller. Note that the corresponding Interrupt Status bit is not affected by the interrupt enable.
9	0h RO	**Fast Back to Back Enable (FBE):**
8	0h RW	**SERR:** When set to 1, the XHC is capable of generating (internally) SERR#.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**Wait Cycle Control (WCC):**
6	0h RW	**Parity Error Response (PER):** When set to 1, the XHCI Host Controller will check for correct parity (on its internal interface) and halt operation when bad parity is detected during the data phase as recommended by the XHCI specification. Note that this applies to both requests and completions from the system interface. This bit must be set in order for the parity errors to generate SERR#.
5	0h RO	**VGA Palette Snoop (VPS):**
4	0h RO	**Memory Write Invalidate (MWI):**
3	0h RO	**Special Cycle Enable (SCE):**
2	0h RW	**Bus Master Enable (BME):** When set, it allows XHC to act as a bus master. When cleared, it disable XHC from initiating transactions on the system bus.
1	0h RW	**Memory Space Enable (MSE):** This bit controls access to the XHC Memory Space registers. If this bit is set, accesses to the XHC registers are enabled. The Base Address register for the XHC should be programmed before this bit is set.
0	0h RO	**I/O Space Enable (IOSE):** Reserved as 0. Read-Only.

16.4 Device Status (STS) — Offset 6h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:21, F:0] + 6h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**Detected Parity Error (DPE):** This bit is set by the SoC whenever a parity error is seen on the internal interface to the XHC host controller, regardless of the setting of bit 6 or bit 8 in the Command register or any other conditions. Software clears this bit by writing a 1 to this bit location.
14	0h RW/1C	**Signaled System Error (SSE):** This bit is set by the SoC whenever it signals SERR# (internally). The SERR_EN bit (bit 8 in the Command Register) must be 1 for this bit to be set. Software clears this bit by writing a 1 to this bit location.
13	0h RW/1C	**Received Master-Abort Status (RMA):** This bit is set when XHC, as a master, receives a master-abort status on a memory access. This is treated as a Host Error and halts the DMA engines. Software clears this bit by writing a 1 to this bit location.
12	0h RW/1C	**Received Target Abort Status (RTA):** This bit is set when XHC, as a master, receives a target abort status on a memory access. This is treated as a Host Error and halts the DMA engines. Software clears this bit by writing a 1 to this bit location.
11	0h RW/1C	**Signaled Target-Abort Status (STA):** This bit is used to indicate when the XHC function responds to a cycle with a target abort.
10:9	0h RO	**DEVSEL# Timing Status (DEVT):** This 2-bit field defines the timing for DEVSEL# assertion. Read-Only.

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW/1C	**Master Data Parity Error Detected (MDPED):** This bit is set by the SoC whenever a data parity error is detected on a XHC read completion packet on the internal interface to the XHC host controller and bit 6 of the Command register is set to 1. Software clears this bit by writing a 1 to this bit location.
7	0h RO	**Fast Back-to-Back Capable (FBBC):** Reserved as 1 Read-Only.
6	0h RO	**User Definable Features (UDF):** Reserved as 0. Read-Only.
5	0h RO	**66 MHz Capable (MC):** Reserved as 0. Read-Only.
4	0h RO	**Capabilities List (CL):** Hardwired to 1 indicating that offset 34h contains a valid capabilities pointer.
3	0h RO/V	**Interrupt Status (IS):** This read-only bit reflects the state of this function's interrupt at the input of the enable/disable logic. This bit is a 1 when the interrupt is asserted. This bit will be 0 when the interrupt is deasserted. The value reported in this bit is independent of the value in the Interrupt Enable bit.
2:0	0h RO	**Reserved (RSVD):** Reserved

16.5 Revision ID (RID) — Offset 8h

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:21, F:0] + 8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO/V	**Revision ID (RID):** See Chap 6 for value.

16.6 Programming Interface (PI) — Offset 9h

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:21, F:0] + 9h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Programming Interface (PI):** A value of 30h indicates that this USB Host Controller conforms to the XHCI specification.

16.7 Sub Class Code (SCC) — Offset Ah

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:21, F:0] + Ah	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Sub Class Code (SCC):** A value of 03h indicates that this is a Universal Serial Bus Host Controller.

16.8 Base Class Code (BCC) — Offset Bh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:21, F:0] + Bh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Base Class Code (BCC):** A value of 0Ch indicates that this is a Serial Bus controller.

16.9 Master Latency Timer (MLT) — Offset Dh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:21, F:0] + Dh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Master Latency Timer (MLT):** Because the XHC controller is internally implemented with arbitration on an internal interface, it does not need a master latency timer. The bits will be fixed at 0.

16.10 Header Type (HT) — Offset Eh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:21, F:0] + Eh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**Multi-Function Bit (MFB):** Read only indicating single function device.

Bit Range	Default & Access	Field Name (ID): Description
6:0	0h RO	**Configuration layout (CL):** Hardwired to 0 to indicate a standard PCI configuration layout.

16.11 Memory Base Address (MBAR) — Offset 10h

Value in this register will be different after the enumeration process.

Type	Size	Offset	Default
CFG	64 bit	[B:0, D:21, F:0] + 10h	0 h

Bit Range	Default & Access	Field Name (ID): Description
63:16	0h RW	**BA:** Bits (63:16) correspond to memory address signals (63:16), respectively. This gives 64 KB of relocatable memory space aligned to 64 KB boundaries.
15:4	0h RO	**Reserved (RSVD):** Reserved
3	0h RO	**PREFETCHABLE:** This bit is hardwired to 0 indicating that this range should not be prefetched.
2:1	0h RO	**TYPE:** If this field is hardwired to 00 it indicates that this range can be mapped anywhere within 32-bit address space. If this field is hardwired to 10 it indicates that this range can be mapped anywhere within 64-bit address space.
0	0h RO	**Resource Type Indicator (RTE):** This bit is hardwired to 0 indicating that the base address field in this register maps to memory space

16.12 USB Subsystem Vendor ID (SSVID) — Offset 2Ch

This register is modified and maintained by BIOS

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:21, F:0] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW/L	**USB Subsystem Vendor ID (SSVID):** This register, in combination with the USB Subsystem ID register, enables the operating system to distinguish each subsystem from the others.

16.13 USB Subsystem ID (SSID) — Offset 2Eh

This register is modified and maintained by BIOS

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:21, F:0] + 2Eh	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW/L	**USB Subsystem ID (SSID):** BIOS sets the value in this register to identify the Subsystem ID. This register, in combination with the Subsystem Vendor ID register, enables the operating system to distinguish each subsystem from other(s).

16.14 Capabilities Pointer (CAP_PTR) — Offset 34h

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:21, F:0] + 34h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Capabilities Pointer (CAP_PTR):** This register points to the starting offset of the capabilities ranges.

16.15 Interrupt Line (ILINE) — Offset 3Ch

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:21, F:0] + 3Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Interrupt Line (ILINE):** This data is not used by the SoC. It is used as a scratchpad register to communicate to software the interrupt line that the interrupt pin is connected to.

16.16 Interrupt Pin (IPIN) — Offset 3Dh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:21, F:0] + 3Dh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW/L	**Interrupt pin (IPIN):** Bits 7:0 reflect the Interrupt Pin assigned to the host controller by the platform (and are hardwired).

16.17 XHC System Bus Configuration 1 (XHCC1) — Offset 40h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:0] + 40h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/O	**Access Control (ACCTRL):** This bit is used by BIOS to lock/unlock lockable bits. When set to '1' the write access to bits locked by this bit is disabled (locked state). When set to '0', the write access to bit locked by this bit is enabled (unlocked state). Writable once after platform reset.
30:25	0h RW	**ECO1:** Reserved
24	0h RW	**Master/Target Abort SERR (RMTASERR):** When set, it allows the out-of-band error reporting from the xHCI Controller to be reported as SERR# (if SERR# reporting is enabled) and thus set the STS.SSE bit.
23	0h RW/C	**Unsupported Request Detected (URD):** Set the HW when xHCI Controller received an unsupported request posted cycle. Cleared by SW when the bit is written with value of '1'.
22	0h RW	**Unsupported Request Report Enable (URRE):** When set this bit allows the URD bit to be reported as SERR# (if SERR# reporting is enabled) and thus set the STS.SSE bit.
21:19	0h RW	**Inactivity Initiated L1 Enable (IIL1E):** If programmed to non-zero, it allows L1 power managed to be enabled after the time-out period specified. 000: Disabled 001: 32 bb_cclk 010: 64 bb_cclk 011: 128 bb_cclk 100: 256 bb_cclk 101: 512 bb_cclk 110: 1024 bb_cclk 111: 131072 bb_cclk
18	0h RW	**XHC Initiated L1 Enable (XHCIL1E):** If set, allow the XHC initiated L1 power management to be enabled.
17	0h RW	**D3 Initiated L1 Enable (D3IL1E):** If set, allow PCI device state D3 initiated L1 power management to be enables.
16:12	0h RW	**Periodic Complete Pre Wake Time (PCPWT):** The value programmed in this field determines how far in advance of the start of the next micro-frame the host controller must de-assert the "Periodic Complete" signal . This allows for platform wake time before the next scheduled periodic transaction. The value programmed in this field represents the # of bytes consumed in the current micro-frame which is required to allow for the periodic complete to de-assert. This allows for a programmable time to cause the periodic complete to de-assert prior to the start of the next micro-frame. Register Format: Bits (16:12) represnets the # of bytes remaining with a 256B granularity. Periodic Complete will de-assert if the bytes consumed in the current micro-frame is less
11	0h RW	**SW Assisted xHC Idle (SWAXHCI):** This bit being set will indicate xHC idleness (through SW means), which must be a '1' to allow L1 entry, and subsequently allow backbone clock to be gated. This bit is to be set by Intel xHCI driver after checking that the xHCI Controller will stay in idle state for a significant period of time, e.g. all ports disconnected. This bit can be cleared under the following conditions (see SWAXHCI Policy bits in xHC System Bus Configuration 2 register): n SW: SW could write 0 to clear this bit. n HW: HW, under policy control, will clear this bit on an MMIO access to the Host Controller. n HW: HW, under policy control, will clear this bit when HW exits Idle state.
10:8	0h RW	**L23 to Host Reset Acknowledge Wait Count (L23HRAWC):** If programmed to non zero, it allows a wait period after the L23 PHY has shutdown before returning host reset acknowledge to PMC. 000: Disabled 001: 128 bb_cclk 010: 256 bb_cclk 011: 512 bb_cclk 100: 1024 bb_cclk 101: 2048 bb_cclk 110: 4096 bb_cclk 111: 131072 bb_cclk
7:6	0h RW	**Upstream Type Arbiter Grant Count Posted (UTAGCP):** Grant count for IOSF upstream L2 request type arbiter for posted type
5:4	0h RW	**Upstream Type Arbiter Grant Count Non Posted (UDAGCNP):** Grant count for IOSF upstream L2 type arbiter for non-posted type

Bit Range	Default & Access	Field Name (ID): Description
3:2	0h RW	**Upstream Type Arbiter Grant Count Completion (UDAGCCP) (UDAGCCP):** Grant count for IOSF upstream L2 type arbiter for completion type
1:0	0h RW	**Upstream Device Arbiter Grant Count (UDAGC) (UDAGC):** Grant count for IOSF upstream L1 device arbiter

16.18 XHC System Bus Configuration 2 (XHCC2) — Offset 44h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:0] + 44h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**OC Configuration Done (OCCFGDONE):** This bit is used by BIOS to prevent spurious switching during OC configuration. It must be set by BIOS after configuration of the OC mapping bits is complete. Once this bit is set, OC mapping shall not be changed by SW.
30:26	0h RW	**ECO1:** Reserved
25	0h RW	**DMA Request Boundary Crossing Control (DREQBCC):** This bit controls the boundary crossing limit of each Read/Write Request. 0: 4KB 1: 64B
24:22	0h RW	**IDMA Read Request Size Control (IDMA_RDREQSZCTRL):** Read Request Size Control: This bit controls the maximum size of each Read Request. 000: 128B 001: 256B 011 - 110: Reserved 111: 64B
21	0h RW	**XHC Upstream Read Relaxed Ordering Enable (XHCUPRDROE):** This policy controls the Relaxed Ordering attribute for upstream reads. 0 - xHC will clear RO for all upstream read requests. 1 - xHC will set RO for all upstream read requests.
20	0h RW	**IOSF Sideband Register Access Disable (IOSFSRAD):** When set, it disables the IOSF sideband interface from accepting any host space register access.
19:14	0h RW	**Upstream Non-Posted Pre-Allocation (UNPPA):** This field reserves data sizes, in 64 byte chunks, of the downstream completion resource. This value is zero based. 000000 - 111111: Pre-allocate 64 bytes - 4096 bytes If set greater than the default allows over-allocation If set less than default allows under-allocation Only allowed to be programmed when BME = 0 and no outstanding downstream completion
13:12	0h RW	**SW Assisted xHC Idle Policy (SWAXHCIP):** Note: Irrespective of the setting of this field, SW write of 0 to SWAXHCI will clear the bit. 00b (default): xHC HW clears SWAXHCI bit upon: n MMIO access to Host Controller OR n xHC HW exits Idle state 01b: xHC HW does not autonomously clear SWAXHCI bit. The bit could be cleared only by SW. 10b: xHC HW clears SWAXHCI upon MMIO acces to Host Controller. xHC HW exit from Idle state will not clear SWAXHCI. 11b: Reserved
11	0h RW	**MMIO Read After MMIO Write Delay Disable (RAWDD):** This field controls delay on MMIO Read after MMIO Write. 0b (Default): Delay MMIO Read after MMIO Write 1b: Do not delay MMIO Read after MMIO Write Note that this delay applies after the second of the two DW writes in the case where the IOSF Gasket splits a QW write into two single DW writes to the IP.
10	0h RW	**MMIO Write After MMIO Write Delay Enable (WAWDE):** This field controls delay on MMIO Write after previous MMIO Write. 0b (Default): Do not delay MMIO Write after previous MMIO Write 1b: Delay MMIO Write after previous MMIO Write Note that the delay count does not apply on the second of the two DW writes that are generated by IOSF Gasket when it splits a QW write into two. In other words, the second of the two DW writes could happen without any delay with respect to the first DW write. This choice is being made for ease of ECO. The delay count, in this case, will apply after the second of the two DW writes.

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW	**SW Assisted Cx Inhibit (SWACXIHB):** This field controls how the DMI L1 inhibit signal from USB3 to PMC will behave. 00: Never inhibit Cx 01: Inhibit Cx when Isochronous Endpoint is active (PPT Behavior) 10: Inhibit Cx when Periodic Active as defined in 40.4.3.2.1 11: Always inhibit Cx
7:6	0h RW	**SW Assisted DMI L1 Inhibit (SWADMIL1IHB):** This field controls how the DMI L1 inhibit signal from USB3 to DMI will behave. 00: Never inhibit DMI L1. 01: Inhibit DMI L1 when Isochronous Endpoint is active (PPT Behavior). 10: Inhibit DMI L1 when Priodic Active as defined in 40.4.3.2.1. 11: Inhibit DMI L1 if XHCC1.SWAXHCI = 0.
5:3	0h RW	**L1 Force P2 Clock Gating Wait Count (L1FP2CGWC):** If programmed to non zero, it allows L1 force P2 gating off the clock to be delayed after the time-out period specified. If wake up event is detected before the time-out, pclk remains alive and trigger L1 exit as though CPU host is causing the wake, 000: Disabled 001: 128 bb_cclk 010: 256 bb_cclk 011: 512 bb_cclk 100: 1024 bb_cclk 101: 2048 bb_cclk 110: 4096 bb_cclk 111: 131072 bb_cclk
2:0	0h RW	**Read Request Size Control (RDREQSZCTRL):** Read Request Size Control: This bit controls the maximum size of each Read Request. 000: 128B 001: 256B 010: 512B 011 - 110: Reserved 111: 64B

16.19 Clock Gating (XHCLKGTEN) — Offset 50h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:0] + 50h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**RSVD2:** Reserved
28	0h RW	**Nak'ing USB2.0 EPs for Backbone Clock Gating and PLL Shutdown (NUEFBCGPS):** This field controls whether Naking USB2.0 EPs, once in Naking low priority schedule, should be considered as active for the considerations for backbone clock gating and PLL shutdown or not. 0: Naking USB2.0 EPs are not considered to be active for Backbone Clock Gating and PLL Shutdown evaluation. 1: Naking USB2.0 EPs are considered to be active for Backbone clock gating and PLL shutdown evaluation.
27	0h RW	**SRAM Power Gate Enable (SRAMPGTEN):** This register enables the SRAM Power Gating when PLL shutdown conditions for all clock domains have been met 0 - Disallow SRAM Power Gating. 1 - Allow SRAM Power Gating
26	0h RW	**SS Link PLL Shutdown Enable (SSLSE):** This register enables the SS P3 state to be exposed to PXP PLL Shutdown conditions on behalf of all USB SS Ports ontop of trunk clock gating. 0 - P3 state NOT allowed to result in PXP PLL shutdown. 1- P3 state allowed to result in PXP PLL shutdown
25	0h RW	**USB2 PLL Shutdown Enable (USB2PLLSE):** When set, this bit allows USB2 PLL to be shutdown when HS Link trunk clock is gated, and xHC can tolerate PLL spin up time for subsequent clock request. Note: if USB2 PLL Shutdown Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value.
24	0h RW	**IOSF Sideband Trunk Clock Gating Enable (IOSFSTCGE):** When set, this bit allows the IOSF sideband clock trunk to be gated when idle conditions are met.
23:20	0h RW	**HS Backbone PXP Trunk Clock Gate Enable (HSTCGE):** This register determines the HS Ux state(s) which will be exposed to Backbone PXP trunk gating of core clock. Uy is a state allowed to result in trunk gating when ss_tcg_ux_en(x) is asserted. (0) ==) U0 or deeper (1) ==) NA (no support for U1) (2) ==) U2 (L1) or deeper (3) ==) U3 (L2) or deeper

Bit Range	Default & Access	Field Name (ID): Description
19:16	0h RW	**SS Backbone PXP Trunk Clock Gate Enable (SSTCGE):** This register determines the SS Ux state(s) which will be exposed to Backbone PXP trunk gating of core clock. Uy is a state allowed to result in trunk gating when ss_tcg_ux_en(x) is asserted. (0) ==) U0 or deeper (1) ==) U1 or deeper (2) ==) U2 or deeper (3) ==) U3 or deeper
15	0h RW	**XHC Ignore_EU3S (XHCIGEU3S):** This register determines if the xHC will use the EU3S as a condition to allow for Frame timer gating. 0 - xHC may allow frame timer to be gated when EU3S is set and all ports are in the required state. 1 - xHC may allow frame timer to be gated regardless of EU3S.
14	0h RW	**XHC Frame Timer Clock Shutdown Enable (XHCFTCLKSE):** This register determines if the xHC will allow the frame timer clock to be gated. 0 - xHC will not allow ICC PLL 96MHz output to be shutdown thus keeping the frame timer running. 1 - xHC will allow ICC PLL 96MHz output to be shutdown under specific conditions.
13	0h RW	**XHC Backbone PXP Trunk Clock Gate In Presence of ISOCH EP (XHCBBTCGIPISO):** This register controls the policy on allowing Backbone PXP trunk clock gate in the presence of IDLE ISOCH EP s with active DB. Allows the periodic active to be used in enabling Backbone PXP trunk clock gating of core clock. 0 Trunk gate of core clock is not allowed when ISOCH EP DB is set and ISOCH EP s are idle. 1 Allow trunk gate of core clock when ISOCH EP DB is set and ISOCH EP s are idle.
12	0h RW	**XHC HS Backbone PXP Trunk Clock Gate U2 non RWE (XHCHSTCGU2NRWE):** This register controls the policy on allowing Backbone PXP trunk gating of core clock when there is atleast 1 non Remote Wake Enabled HS Port in U2. 0 Prevent trunk gate of core clock when a non RWE HS Port is in U2. 1 Allow trunk gate of core clock when a non RWE HS Port is in U2.
11:10	0h RW	**XHC USB2 PLL Shutdown Lx Enable (XHCUSB2PLLSDLE):** This register determines the HS Link state(s) which will be exposed to USB2 PLL Shutdown conditions on behalf of all USB2 HS Ports. Ly is a state allowed to result in USB2 PLL Shutdown when en(x) is asserted. (0) ==) L1 or deeper (1) ==) L2 or deeper
9:8	0h RW	**HS Backbone PXP PLL Shutdown Ux Enable (HSUXDMIPLLSE):** This register determines the Ux state(s) which will be exposed to PXP PLL Shutdown conditions. PLL Shutdown is allowed in: 00b Disabled (Link states shall be disabled for DMI PLL shutdown) 01b U0 or conditions for 10b setting. 10b U2 or conditions for 11b setting. 10b U3, Disconnected, Disabled or Powered-Off.
7:5	0h RW	**SS Backbone PXP PLL Shutdown Ux Enable (SSPLLSUE):** This register determines the Ux state(s) which will be exposed to DMI PLL Shutdown conditions. PLL Shutdown is allowed in: 000b Disabled (Link states shall be ignored for DMI PLL shutdown). 001b U0 or conditions for 010b setting 010b U1 or conditions for 011b setting 011b U2 or conditions for 100b setting 100b U3, Disconnected, Disabled or Powered-Off
4	0h RW	**XHC Backbone Local Clock Gating Enable (XHCBLCGE):** When set, this bit allows XHCI Controller IP backbone clock to be locally gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value.
3	0h RW	**HS Link Trunk Clock Gating Enable (HSLTCGE):** When set, this bit allows High Speed Link control's 480 MHz and its 48/60 MHz link clock trunk to be gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value. BIOS need to query fuses to see if HW is enabled.
2	0h RW	**SS Link Trunk Clock Gating Enable (SSLTCGE):** When set, this bit allows the SuperSpeed Link control's 250 MHz and its divided 125 MHz link clock trunk to ge gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value. BIOS need to query fuses to see if HW is enabled.
1	0h RW	**IOSF Backbone Trunk Clock Gating Enable (IOSFBTCGE):** When set, this bit allows the IOSF backbone clock trunk to be gated when idle conditions are met.
0	0h RW	**IOSF Gasket Backbone Local Clock Gating Enable (IOSFBLCGE):** When set, this bit allows the IOSF Gasket backbone clock to be locally gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value. BIOS need to query fuses to see if HW is enabled.

16.20 Audio Time Synchronization (AUDSYNC) — Offset 58h

This 32 bit register is used for audio stream synchronization across different devices. Global signal sample_now captures a value in AUDSYNC register.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:0] + 58h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**RSVD2:** Reserved
29:16	0h RO/V	**Captured Frame List Current Index/Frame Number (CMFI):** The value in this register is updated in response to sample_now signal. Bits (29:16) reflect state of bits (13:0) of FRINDEX
15:13	0h RO	**RSVD1:** Reserved
12:0	0h RO/V	**Captured Micro-frame BLIF (CMFB):** The value is updated in response to sample_now signal and provides information about offset within micro-frame. Captured value represents number of 8 high-speed bit time units from start of micro-frame. At the beginning of micro-frame captured value will be 0 and increase to maximum value at the end. Default maximum value is 7499 but it may be changed as result of adjustment done in FLA.

16.21 Serial Bus Release Number (SBRN) — Offset 60h

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:21, F:0] + 60h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Serial Bus Release Number (SBRN):** A value of 30h indicates that this controller follows USB release 3.0.

16.22 Frame Length Adjustment (FLADJ) — Offset 61h

This feature is used to adjust any offset from the clock source that generates the clock that drives the SOF counter. When a new value is written into these six bits, the length of the frame is adjusted. Its initial programmed value is system dependent based on the accuracy of hardware USB clock and is initialized by system BIOS. This register should only be modified when the HChalted bit in the USBSTS register is a one. Changing value of this register while the host controller is operating yields undefined results.

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:21, F:0] + 61h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**Reserved (RSVD):** Reserved
6	0h RO	**No Frame Length Timing Capability (NO_FRAME_LENGTH_TIMING_CAP):** This flag is set to 1 to indicate that the host controller does not support a programmable Frame Length Timing Value field.
5:0	0h RO	**Frame Length Timing Value (FLTV):** SOF micro-frame length) is equal to 59488 + value in this field. The default value is decimal 32 (20h), which gives a SOF cycle time of 60000. Frame Length (number of High Speed bit times) FLADJ Value (decimal) (decimal) 59488 0 (00h) 59504 1 (01h) 59520 2 (02h) ... 59984 31 (1Fh) 60000 32 (20h) ... 60480 62 (3Eh) 60496 63 (3Fh) Each decimal value change to this register corresponds to 16 high-speed bit times. The SOF cycle time (number of SOF counter clock periods to generate a SOF micro-frame length) is equal to 59488 + value in this field. The default value is decimal 32 (20h), which gives a SOF cycle time of 60000. Frame Length (# High Speed bit times) FLADJ Value

16.23 Best Effort Service Latency (BESL) — Offset 62h

Bset Effort Service Latency.

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:21, F:0] + 62h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:4	0h RW/L	**Default Best Effort Service Latency Deep (DBESLD):** Default Best Effort Service Latency (DBESLD) If the value of this field is non-zero, it defines the recommended value for programming the PORTPMSC register BESLD field. This is programmed by BIOS based on platform parameters.
3:0	0h RW/L	**Default Best Effort Service Latency (DBESL):** If the value of this field is non-zero, it defines the recommended value for programming the PORTPMSC register BESL field. This is programmed by BIOS based on platform parameters.

16.24 PCI Power Management Capability ID (PM_CID) — Offset 70h

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:21, F:0] + 70h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**PCI Power Management Capability ID (PM_CID):** A value of 01h indicates that this is a PCI Power Management capabilities field.

16.25 Next Item Pointer #1 (PM_NEXT) — Offset 71h

This register is modified and maintained by BIOS

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:21, F:0] + 71h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW/L	**Next Item Pointer #1 (PM_NEXT):** This register defaults to 80h, which indicates that the next capability registers begin at configuration offset 80h. This register is writable when the Access Control bit is set to '0'. This allows BIOS to effectively hide the next capability registers, if necessary. This register should only be written during system initialization before the plug-and-play software has enabled any master-initiated traffic. Values of: 80h implies next capability is MSI 00h implies that MSI capability is hidden. Note: This value is never expected to be programmed.

16.26 Power Management Capabilities (PM_CAP) — Offset 72h

Normally, this register is read-only to report capabilities to the power management software. In order to report different power management capabilities depending on the system in which the SoC is used, the write access to this register is controlled by the Access Control bit (ACCTRL). The value written to this register does not affect the hardware other than changing the value returned during a read. This register is modified and maintained by BIOS

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:21, F:0] + 72h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:11	0h RW/L	**PME_SUPPORT:** This 5-bit field indicates the power states in which the function may assert PME#. The SoC XHC does not support the D1 or D2 states. For all other states, the SoC XHC is capable of generating PME#. Software should never need to modify this field.
10	0h RW/L	**D2_SUPPORT:** The D2 state is not supported.
9	0h RW/L	**D1_SUPPORT:** The D1 state is not supported.
8:6	0h RW/L	**AUX_CURRENT:** The SoC XHC reports 375mA maximum Suspend well current required when in the D3cold state. This value can be written by BIOS when a more accurate value is known.
5	0h RW/L	**DSI:** The SoC reports 0, indicating that no device-specific initialization is required.
4	0h RO	**Reserved (RSVD):** Reserved
3	0h RW/L	**PME Clock (PMECLOCK):** The SoC reports 0, indicating that no PCI clock is required to generate PME#.
2:0	0h RW/L	**VERSION:** The SoC reports 010, indicating that it complies with Revision 1.1 of the PCI Power Management Specification.

16.27 Power Management Control/Status (PM_CS) — Offset 74h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:21, F:0] + 74h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**PME_STATUS:** This bit is set when the SoC XHC would normally assert the PME# signal independent of the state of the PME_En bit. Writing a 1 to this bit will clear it and cause the internal PME to deassert (if enabled). Writing a 0 has no effect. This bit must be explicitly cleared by the operating system each time the operating system is loaded.
14:13	0h RO	**DATA_SCALE:** The SoC hardwires these bits to 00 because it does not support the associated Data register.
12:9	0h RO	**DATA_SELECT:** The SoC hardwires these bits to 0000 because it does not support the associated Data register.
8	0h RW	**PME_EN:** A 1 enables the SoC XHC to generate an internal PME signal when PME_Status is 1. This bit must be explicitly cleared by the operating system each time it is initially loaded.
7:4	0h RO	**Reserved (RSVD):** Reserved
3	0h RO	**No Soft Reset (NSR):** No_Soft_Reset - When set ("1"), this bit indicates that devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset. Configuration Context is preserved. Upon transition from the D3hot to the D0 Initialized state, no additional operating system intervention is required to preserve Configuration Context beyond writing the PowerState bits. Transition from D3hot to D0 by a system or bus segment reset will return to the device state D0 Uninitialized with only PME context preserved if PME is supported and enabled.
2	0h RO	**Reserved (RSVD2):** Reserved
1:0	0h RW	**POWERSTATE:** This 2-bit field is used both to determine the current power state of XHC function and to set a new power state. The definition of the field values are: 00b - D0 state 11b - D3hot state If software attempts to write a value of 10b or 01b in to this field, the write operation must complete normally, however, the data is discarded and no state change occurs. When in the D3hot state, the SoC must not accept accesses to the XHC memory range, but the configuration space must still be accessible.

16.28 Message Signaled Interrupt CID (MSI_CID) — Offset 80h

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:21, F:0] + 80h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Capability ID (CID):** Indicates that this is an MSI capability

16.29 Next item pointer (MSI_NEXT) — Offset 81h

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:21, F:0] + 81h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW/L	**NEXT:** Indicates that this is the last item on the capability list

16.30 Message Signaled Interrupt Message Control (MSI_MCTL) — Offset 82h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:21, F:0] + 82h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:9	0h RO	**Reserved (RSVD):** Reserved
8	0h RO	**Per-Vector Masking Capable (PVM):** Specifies whether controller supports MSI per vector masking. Not supported
7	0h RO	**64 Bit Address Capable (C64):** Specifies whether capable of generating 64-bit messages. This device is 64-bit capable.
6:4	0h RW	**Multiple Message Enable (MME):** Indicates the number of messages the controller should assert. This device supports multiple message MSI.
3:1	0h RO	**Multiple Message Capable (MMC):** Indicates the number of messages the controller wishes to assert. This field must be set by HW to reflect the number of Interrupters supported. Encoding number of Vectors requested (number of Interrupters) 000 1 001 2 010 4 011 8 100 16 101 32 110-111 Reserved
0	0h RW	**MSI Enable (MSIE):** If set to 1, MSI is enabled and the traditional interrupt pins are not used to generate interrupts. If cleared to 0, MSI operation is disabled and the traditional interrupt pins are used.

16.31 Message Signaled Interrupt Message Address (MSI_MAD) — Offset 84h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:0] + 84h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**ADDR:** Lower DW of system specified message address, always DWORD aligned
1:0	0h RO	**Reserved (RSVD):** Reserved

16.32 Message Signaled Interrupt Upper Address (MSI_MUAD) — Offset 88h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:0] + 88h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Upper Addr (UPPERADDR):** Upper DW of system specified message address.

16.33 Message Signaled Interrupt Message Data (MSI_MD) — Offset 8Ch

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:21, F:0] + 8Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW	**DATA:** This 16-bit field is programmed by system software if MSI is enabled. Its content is driven onto the lower word (PCI AD(15:0)) during the data phase of the MSI memory write transaction. The Multiple Message Enable field (bits 6-4 of the Message Control register) defines the number of low order message data bits the function is permitted to modify to generate its system software allocated vectors. For example, a Multiple Message Enable encoding of 010 indicates the function has been allocated four vectors and is permitted to modify message data bits 1 and 0 (a function modifies the lower message data bits to generate the allocated number of vectors). If the Multiple Message Enable field is 000, the function is not permitted to modify the message data.

16.34 Device Idle Capability (DEVIDLE) — Offset 90h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:0] + 90h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**VID:**
27:24	0h RO	**REV:**
23:16	0h RO	**LENGTH:** Indicates that this capability is 16 bytes long.
15:8	0h RO	**Next Capability Pointer (NCP):** This field contains the offset to the next PCI Capability structure or 000h if no other items exist in the linked list of Capabilities.
7:0	0h RO	**Capability ID (CID):**

16.35 Vendor Specific Header (VSHDR) — Offset 94h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:0] + 94h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RO	**VSEC Length (VSEC_LENGTH):** This field indicates the number of bytes in the entire VSEC structure, including the PCI Extended Capability header, the Vendor- Specific header, and the Vendor-Specific register
19:16	0h RO	**VSEC Rev (VSEC_REV):** This field is a vendor-defined version number that indicates the version of the VSEC structure. Software must qualify the Vendor ID and VSEC ID before interpreting this field.
15:0	0h RO	**VSEC ID (VSEC_ID):** This field is a vendor-defined ID number that indicates the nature and format of the VSEC structure. Software must qualify the Vendor ID before interpreting this field.

16.36 SW LTR POINTER (SWLTRPTR) — Offset 98h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:0] + 98h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**SW LTR Update MMIO Offset Location (SW_LTR_UPDT_MMIO_OFFSET):** This register contains the location pointer to the SW LTR register in MMIO space, as an offset from the specified BAR. The value of this register is a don't care, if the Valid bit is not set.
3:1	0h RO	**Base Address Register Number (BARNUM):** Contains the 0's based AR Number of the BAR which contains the location of the SW LTR MMIO register. When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number. Bar 0 is the 1st BAR. The value of this register is a don't care, if the Valid bit is not set.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**VALID:** Set to '1' to indicate that the function has implemented a SW LTR register as specified in the DevIdle that can be located using the SWLTRLOC register and BARNUM. Set to '0' to indicate that the function has not implemented a SW LTR register that is compliant to the DevIdle definition. This could be because the function has not implemented SW LTR at all, or has a device specific version of SW LTR.

16.37 Device Idle Pointer Register (DEVIDLEPTR) — Offset 9Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:0] + 9Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RW/L	**DevIdle MMIO Offset Location (DEVIDLELOC):** This register contains the location pointer to the DevIdle register in MMIO space, as an offset from the specified BAR. The value of this register is a don't care, if the Valid bit is not set.
3:1	0h RO	**Base Address Register Number (BARNUM):** Contains the 0's based BAR Number of the BAR which contains the location of the DevIdle MMIO register. When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number. Bar 0 is the 1st BAR. The value of this register is a don't care, if the Valid bit is not set.
0	0h RW/L	**VALID:** Set to '1' to indicate that the function has implemented a DevIdle register as specified in the DevIdle that can be located using the DEVIDLELOC register and BARNUM. Set to '0' to indicate that the function has not implemented a DevIdle register that is compliant to the DevIdle definition. This could be because the function has not implemented DevIdle at all, or has a device specific version of DevIdle.

16.38 Device Idle Power ON Latency (DEVIDLEPOL) — Offset A0h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:21, F:0] + A0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	**RSVD1:** Reserved
12:10	0h RW/L	**Power On Latency Scale (POLS):** Latency Scale multiplier: 010: 1us 011: 32us All other settings are reserved. The value of this register is multiplied with the Power On Latency Value (POLV) to provide the DevIdle power on exit latency multiplier scale from 1us to 32ms. This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value. This register is undefined if the DevIdle.Valid bit is '0'.
9:0	0h RW/L	**Power On Latency Value (POLV):** 10-bit value that is multiplied by the Power On Latency Scale. A value of 0 indicates a power on latency of less than 1us. This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value. This register is undefined if the DevIdle.Valid bit is '0'.

16.39 High Speed Configuration 2 (HSCFG2) — Offset A4h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:0] + A4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**RSVD1:** Reserved
18	0h RW	**PORT1_HOST_MODE_OVERRIDE:** When set, this bit causes the Host_Device mux on port 1 to be forced into the Host mode.
17:16	0h RW	**EUSB2SEL:** The two bits are associate with USB2 ports 1 - bit 16 and 2 - bit 2 0: Port is mapped to USB2 1: Port is mapped to eUSB2
15	0h RW	**HS ASYNC Active IN Mask (HSAAIM):** Determines if the Async Active will mask/ ignore IN EP s. 0 HS ASYNC Active will include IN EP s. 1 HS ASYNC Active will mask/ ignore IN EP s.
14	0h RW	**HS OUT ASYNC Active Polling EP Mask (HSOAAPEPM):** Determines if the Async Active for OUT HS/FS/LS masks/ignores EP s that are polling/PINGing (HS) due to NAK. 0 HS OUT ASYNC Active will include EP s that are polling. 1 HS OUT ASYNC Active will mask/ignore EP s that are polling.
13	0h RW	**HS IN ASYNC Active Polling EP Mask (HSIAAPEPM):** Determines if the Async Active for IN HS/FS/LS masks/ignores EP s that are polling due to NAK. 0 HS IN ASYNC Active will include EP s that are polling. 1 HS IN ASYNC Active will mask/ignore EP s that are polling.
12:11	0h RW	**HS INTR IN Periodic Active Policy Control (HSIIPAPC):** Controls how the HS INTR IN periodic active is used to generate the global periodic active. This will determine how the smallest service interval among active EP s and number of active EP s are used. 0 HS INTR IN periodic active will be used to generate periodic active if Service Interval Threshold OR Numb of EP Threshold values meet the requirement. 1 HS INTR IN periodic active will be used to generate periodic active if Service Interval Threshold AND Numb of EP Threshold values meet the requirement. 2 Always allow HS INTR EP s to be used in the generation of the global Periodic Active indication. 3 Never allow HS INTR EP s to be used in the generation of the global Periodic Active indicaiton
10:4	0h RW	**HS INTR IN Periodic Active Num of EP Threshold (HSIIPANEPT):** Defines the threshold used to determine if Periodic Acive may include HS/FS/LS INTR IN EP active indication. If there are more than NumEPThreshold active HS/FS/LS INTR EP s then they may be included as part of the periodic active generation.
3:0	0h RW	**HS INTR IN Periodic Active Service Interval Threshold (HSIIPASIT):** Defines the Service Interval threshold used to determine if Periodic Acive will include HS/FS/LS INTR IN EP active indication. If there are any active HS/FS/LS INTR EP s with a service interval less than or equal to this threshold then they may be included as part of the periodic active generation.

16.40 XHCI USB2 Overcurrent Pin Mapping 1 (U2OCM1) — Offset B0h

The RW/L property of this register is controlled by OCCFDONE bit.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:0] + B0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**RSVD:** Reserved
8:0	0h RW/L	**OC Mapping (OCM):**

16.41 XHCI USB2 Overcurrent Pin Mapping 2 (U2OCM2) — Offset B4h

Reserved

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:0] + B4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**RSVD:** Reserved
8:0	0h RW/L	**OC Mapping (OCM):**

16.42 XHCI USB3 Overcurrent Pin Mapping 1 (U3OCM1) — Offset D0h

The RW/L property of this register is controlled by OCCFDONE bit.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:0] + D0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**RSVD:** Reserved
6:0	0h RW/L	**OC Mapping (OCM)**

16.43 XHCI USB3 Overcurrent Pin Mapping 2 (U3OCM2) — Offset D4h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:0] + D4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**RSVD:** Reserved
6:0	0h RW/L	**OC Mapping (OCM):**

16.44 XHCC3 — Offset FCh

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:0] + FCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RSVD1):** Reserved
3	0h RW	**Error Handling : Disable Command Parity Check (DISABLE_COMMAND_PARITY_CHECK):** When set to 1, XHCI Host Controller disables checking of Command Parity on command received as a target on its IOSF Primary interface
2	0h RW	**Error Handling : Disable Data Parity Check (DISABLE_DATA_PARITY_CHECK):** When set to 1, XHCI Host Controller disables checking of Data Parity on data received as a target on its IOSF Primary interface
1	0h RW	**Error Handling : Enable ECC Error Response (ENABLE_ECC_ERROR_RESPONSE):** When set to 1, XHCI Host Controller will check for ECC on RFs (that support ECC) and halt operation when uncorrectable ECC is detected
0	0h RW/L	**Function Disable (FXN_DISABLE):** When set will disable the xHC from being operational.

16.45 Capability Registers Length (CAPLENGTH)—Offset 0h

This register is modified and maintained by BIOS

Access Method

Type: MEM Register (Size: 8 bits)

Device:

Function:

Default: 80h

Bit Range	Default & Access	Field Name (ID): Description
7:0	80h RW/L	**Capability Registers Length (CAPLENGTH)**

16.46 Host Controller Interface Version Number (HCIVERSION)—Offset 2h

This register is modified and maintained by BIOS

Access Method

Type: MEM Register (Size: 16 bits)

Device:
Function:

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
15:0	100h RW/L	**Host Controller Interface Version Number (HCIVERSION)**

16.47 Structural Parameters 1 (HCSPARAMS1)—Offset 4h

This register is modified and maintained by BIOS

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 10000820h

Bit Range	Default & Access	Field Name (ID): Description
31:24	10h RW/L	**Number of Ports (MaxPorts)**
23:19	0h RW/L	**Rsvd1 (Rsvd1)**
18:8	8h RW/L	**Number of Interrupters (MaxIntrs)**
7:0	20h RW/L	**Number of Device Slots (MaxSlots)**

16.48 Structural Parameters 2 (HCSPARAMS2)—Offset 8h

This register is modified and maintained by BIOS

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 94000054h

Bit Range	Default & Access	Field Name (ID): Description
31:27	12h RW/L	**Max Scratchpad Buffers LO (MaxScratchpadBufs)**
26	1h RW/L	**Scratchpad Restore (SPR)**

Bit Range	Default & Access	Field Name (ID): Description
25:21	0h RW/L	**Max Scratchpad Buffers HI (MaxScratchpadBufs_HI)**
20:8	0h RW/L	**Rsvd1 (Rsvd1)**
7:4	5h RW/L	**Event Ring Segment Table Max (ERSTMax)**
3:0	4h RW/L	**Isochronous Scheduling Threshold (IST)**

16.49 Structural Parameters 3 (HCSPARAMS3)—Offset Ch

This register is modified and maintained by BIOS

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 40001h

Bit Range	Default & Access	Field Name (ID): Description
31:16	4h RW/L	**U2 Device Exit Latency (U2DEL)**
15:8	0h RW/L	**Rsvd1 (Rsvd1)**
7:0	1h RW/L	**U1 Device Exit Latency (U1DEL)**

16.50 Capability Parameters (HCCPARAMS)—Offset 10h

This register is modified and maintained by BIOS

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 200077C1h

Bit Range	Default & Access	Field Name (ID): Description
31:16	2000h RW/L	**xHCI Extended Capabilities Pointer (xECP)**
15:12	7h RW/L	**Maximum Primary Stream Array Size (MaxPSASize)**
11	0h RW/L	**Contiguous Frame ID Capability (CFC)**

Bit Range	Default & Access	Field Name (ID): Description
10	1h RW/L	**Stopped EDLTA Capabilty (SEC)**
9	1h RW/L	**Stopped - Short Packet Capability (SPC)**
8	1h RW/L	**Parst All Event Data (PAE)**
7	1h RW/L	**No Secondary SID Support (NSS)**
6	1h RW/L	**Latency Tolerance Messaging Capability (LTC)**
5	0h RW/L	**Light HC Reset Capability (LHRC)**
4	0h RW/L	**Port Indicators (PIND)**
3	0h RW/L	**Port Power Control (PPC)**
2	0h RW/L	**Context Size (CSZ)**
1	0h RW/L	**BW Negotiation Capability (BNC)**
0	1h RW/L	**64-bit Addressing Capability (AC64)**

16.51 Doorbell Offset (DBOFF)—Offset 14h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:2	C00h RO	**Doorbell Array Offset (DBAO)**
1:0	0h RO	**Rsvd1 (Rsvd1)**

16.52 Runtime Register Space Offset (RTSOFF)—Offset 18h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2000h

Bit Range	Default & Access	Field Name (ID): Description
31:5	100h RO	**Runtime Register Space Offset (RTRSO)**
4:0	0h RO	**Rsvd1 (Rsvd1)**

16.53 USB Command (USBCMD)—Offset 80h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Rsvd2 (Rsvd2)**
11	0h RW	**Enable U3 MFINDEX Stop (EU3S)**
10	0h RW	**Enable Wrap Event (EWE)**
9	0h RW	**Controller Restore State (CRS)**
8	0h RW	**Controller Save State (CSS)**
7	0h RW	**Light Host Controller Reset (LHCRST)**
6:4	0h RO	**Rsvd1 (Rsvd1)**
3	0h RW	**Host System Error Enable (HSEE)**
2	0h RW	**Interrupter Enable (INTE)**
1	0h RW	**Host Controller Reset (HCRST)**
0	0h RW	**Run/Stop (RS)**

16.54 USB Status (USBSTS)—Offset 84h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
31:13	0h RO	**Rsvd3 (Rsvd3)**
12	0h RO	**Host Controller Error (HCE):** This bit is not preset in HC, this is deviation from XHCI 1.0 spec.
11	0h RO	**Controller Not Ready (CNR):** This is deviation from XHCI 1.0 spec.
10	0h RW/C	**Save/Restore Error (SRE)**
9	0h RO	**Restore State Status (RSS)**
8	0h RO	**Save State Status (SSS)**
7:5	0h RO	**Rsvd2 (Rsvd2)**
4	0h RW/C	**Port Change Detect (PCD)**
3	0h RW/C	**Event Interrupt (EINT)**
2	0h RW/C	**Host System Error (HSE)**
1	0h RO	**Rsvd1 (Rsvd1)**
0	1h RO	**HCHalted (HCH)**

16.55 Page Size (PAGESIZE)—Offset 88h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	1h RO	**Page Size (PAGESIZE)**

16.56 Device Notification Control (DNCTRL)—Offset 94h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RW	**Notification Enable (N0_N15)**

16.57 Command Ring Low (CRCR_LO)—Offset 98h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Command Ring Pointer (CRP)**
5:4	0h RO	**Rsvd1 (Rsvd1)**
3	0h RO	**Command Ring Running (CRR)**
2	0h RW/1S	**Command Abort (CA)**
1	0h RW/1S	**Command Stop (CS)**
0	0h RW	**Ring Cycle State (RCS)**

16.58 Command Ring High (CRCR_HI)—Offset 9Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Command Ring Pointer (CRP)**

16.59 Device Context Base Address Array Pointer Low (DCBAAP_LO)—Offset B0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Device Context Base Address Array Pointer (DCBAAP)**
5:0	0h RO	**Rsvd1 (Rsvd1)**

16.60 Device Context Base Address Array Pointer High (DCBAAP_HI)—Offset B4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Device Context Base Address Array Pointer (DCBAAP)**

16.61 Configure (CONFIG)—Offset B8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**Max Device Slots Enabled (MaxSlotsEn)**

16.62 Port Status and Control USB2 (PORTSC1)—Offset 480h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.63 Port Power Management Status and Control USB2 (PORTPMSC1)—Offset 484h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Port Test Control (PTC):** Note: This register is sticky.
27:17	0h RO	**Rsvd1 (Rsvd1)**
16	0h RW	**Hardware LPM Enable (HLE)**
15:8	0h RW	**Device Address (DA):** Note: This register is sticky.
7:4	0h RW	**Host Initiated Resume Duration (HIRD):** Note: This register is sticky.
3	0h RW	**Remote Wake Enable (RWE):** Note: This register is sticky.
2:0	0h RW	**L1 Status (L1S):** Note: This register is sticky.

16.64 Port X Hardware LPM Control Register (PORTHLPMC1)—Offset 48Ch

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters neccessary for xHC to automatically generate an LPM Token to the downstream device.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**RESERVED (RSVD)**
13:10	0h RW	**Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1. The HIRDD value is is encoded as follows: 0h: 50 us (default) 1h: 125 us 2h: 200 us ... Fh: 1.175ms The value of 0h is interpreted as 50 us. Each incrementing value adds 75 us to the previous value.
9:2	0h RW	**L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer. This field shall be set to 00h by assertion of PR to '1'. Following are permissible values: 00h: 128 us (default) 01h: 256 us. ... FFh: 65,280us Note: This register is sticky.
1:0	0h RW	**Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used. Following are permissible values: 0: Initiate L1 using HIRD only time out (default) 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD 2,3: Reserved Note: This register is sticky.

16.65 Port Status and Control USB2 (PORTSC2)—Offset 490h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.66 Port Power Management Status and Control USB2 (PORTPMSC2)—Offset 494h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Port Test Control (PTC):** Note: This register is sticky.
27:17	0h RO	**Rsvd1 (Rsvd1)**

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW	**Hardware LPM Enable (HLE)**
15:8	0h RW	**Device Address (DA):** Note: This register is sticky.
7:4	0h RW	**Host Initiated Resume Duration (HIRD):** Note: This register is sticky.
3	0h RW	**Remote Wake Enable (RWE):** Note: This register is sticky.
2:0	0h RW	**L1 Status (L1S):** Note: This register is sticky.

16.67 Port X Hardware LPM Control Register (PORTHLPMC2)—Offset 49Ch

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters neccessary for xHC to automatically generate an LPM Token to the downstream device.

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**RESERVED (RSVD)**
13:10	0h RW	**Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1. The HIRDD value is is encoded as follows: 0h: 50 us (default) 1h: 125 us 2h: 200 us ... Fh: 1.175ms The value of 0h is interpreted as 50 us. Each incrementing value adds 75 us to the previous value.
9:2	0h RW	**L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer. This field shall be set to 00h by assertion of PR to '1'. Following are permissible values: 00h: 128 us (default) 01h: 256 us. ... FFh: 65,280us Note: This register is sticky.
1:0	0h RW	**Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used. Following are permissible values: 0: Initiate L1 using HIRD only time out (default) 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD 2,3: Reserved Note: This register is sticky.

16.68 Port Status and Control USB2 (PORTSC3)—Offset 4A0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.69 Port Power Management Status and Control USB2 (PORTPMSC3)—Offset 4A4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Port Test Control (PTC):** Note: This register is sticky.
27:17	0h RO	**Rsvd1 (Rsvd1)**
16	0h RW	**Hardware LPM Enable (HLE)**
15:8	0h RW	**Device Address (DA):** Note: This register is sticky.
7:4	0h RW	**Host Initiated Resume Duration (HIRD):** Note: This register is sticky.
3	0h RW	**Remote Wake Enable (RWE):** Note: This register is sticky.
2:0	0h RW	**L1 Status (L1S):** Note: This register is sticky.

16.70 Port X Hardware LPM Control Register (PORTHLPMC3)—Offset 4ACh

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters neccessary for xHC to automatically generate an LPM Token to the downstream device.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**RESERVED (RSVD)**
13:10	0h RW	**Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1. The HIRDD value is is encoded as follows: 0h: 50 us (default) 1h: 125 us 2h: 200 us ... Fh: 1.175ms The value of 0h is interpreted as 50 us. Each incrementing value adds 75 us to the previous value.
9:2	0h RW	**L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer. This field shall be set to 00h by assertion of PR to '1'. Following are permissible values: 00h: 128 us (default) 01h: 256 us. ... FFh: 65,280us Note: This register is sticky.
1:0	0h RW	**Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used. Following are permissible values: 0: Initiate L1 using HIRD only time out (default) 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD 2,3: Reserved Note: This register is sticky.

16.71 Port Status and Control USB2 (PORTSC4)—Offset 4B0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.72 Port Power Management Status and Control USB2 (PORTPMSC4)—Offset 4B4h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Port Test Control (PTC):** Note: This register is sticky.
27:17	0h RO	**Rsvd1 (Rsvd1)**
16	0h RW	**Hardware LPM Enable (HLE)**
15:8	0h RW	**Device Address (DA):** Note: This register is sticky.
7:4	0h RW	**Host Initiated Resume Duration (HIRD):** Note: This register is sticky.
3	0h RW	**Remote Wake Enable (RWE):** Note: This register is sticky.
2:0	0h RW	**L1 Status (L1S):** Note: This register is sticky.

16.73 Port X Hardware LPM Control Register (PORTHLPMC4)—Offset 4BCh

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters neccessary for xHC to automatically generate an LPM Token to the downstream device.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**RESERVED (RSVD)**
13:10	0h RW	**Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1. The HIRDD value is is encoded as follows: 0h: 50 us (default) 1h: 125 us 2h: 200 us ... Fh: 1.175ms The value of 0h is interpreted as 50 us. Each incrementing value adds 75 us to the previous value.
9:2	0h RW	**L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer. This field shall be set to 00h by assertion of PR to '1'. Following are permissible values: 00h: 128 us (default) 01h: 256 us. ... FFh: 65,280us Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used. Following are permissible values: 0: Initiate L1 using HIRD only time out (default) 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD 2,3: Reserved Note: This register is sticky.

16.74 Port Status and Control USB2 (PORTSC5)—Offset 4C0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.75 Port Power Management Status and Control USB2 (PORTPMSC5)—Offset 4C4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Port Test Control (PTC):** Note: This register is sticky.
27:17	0h RO	**Rsvd1 (Rsvd1)**
16	0h RW	**Hardware LPM Enable (HLE)**
15:8	0h RW	**Device Address (DA):** Note: This register is sticky.
7:4	0h RW	**Host Initiated Resume Duration (HIRD):** Note: This register is sticky.
3	0h RW	**Remote Wake Enable (RWE):** Note: This register is sticky.
2:0	0h RW	**L1 Status (L1S):** Note: This register is sticky.

16.76 Port X Hardware LPM Control Register (PORTHLPMC5)—Offset 4CCh

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters neccessary for xHC to automatically generate an LPM Token to the downstream device.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**RESERVED (RSVD)**
13:10	0h RW	**Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1. The HIRDD value is is encoded as follows: 0h: 50 us (default) 1h: 125 us 2h: 200 us ... Fh: 1.175ms The value of 0h is interpreted as 50 us. Each incrementing value adds 75 us to the previous value.
9:2	0h RW	**L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer. This field shall be set to 00h by assertion of PR to '1'. Following are permissible values: 00h: 128 us (default) 01h: 256 us. ... FFh: 65,280us Note: This register is sticky.
1:0	0h RW	**Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used. Following are permissible values: 0: Initiate L1 using HIRD only time out (default) 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD 2,3: Reserved Note: This register is sticky.

16.77 Port Status and Control USB2 (PORTSC6)—Offset 4D0h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**

Bit Range	Default & Access	Field Name (ID): Description
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.78 Port Power Management Status and Control USB2 (PORTPMSC6)—Offset 4D4h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Port Test Control (PTC):** Note: This register is sticky.
27:17	0h RO	**Rsvd1 (Rsvd1)**
16	0h RW	**Hardware LPM Enable (HLE)**
15:8	0h RW	**Device Address (DA):** Note: This register is sticky.
7:4	0h RW	**Host Initiated Resume Duration (HIRD):** Note: This register is sticky.
3	0h RW	**Remote Wake Enable (RWE):** Note: This register is sticky.
2:0	0h RW	**L1 Status (L1S):** Note: This register is sticky.

16.79 Port X Hardware LPM Control Register (PORTHLPMC6)—Offset 4DCh

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters neccessary for xHC to automatically generate an LPM Token to the downstream device.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**RESERVED (RSVD)**

Bit Range	Default & Access	Field Name (ID): Description
13:10	0h RW	**Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1. The HIRDD value is is encoded as follows: 0h: 50 us (default) 1h: 125 us 2h: 200 us ... Fh: 1.175ms The value of 0h is interpreted as 50 us. Each incrementing value adds 75 us to the previous value.
9:2	0h RW	**L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer. This field shall be set to 00h by assertion of PR to '1'. Following are permissible values: 00h: 128 us (default) 01h: 256 us. ... FFh: 65,280us Note: This register is sticky.
1:0	0h RW	**Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used. Following are permissible values: 0: Initiate L1 using HIRD only time out (default) 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD 2,3: Reserved Note: This register is sticky.

16.80 Port Status and Control USB2 (PORTSC7)—Offset 4E0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.81 Port Power Management Status and Control USB2 (PORTPMSC7)—Offset 4E4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Port Test Control (PTC):** Note: This register is sticky.
27:17	0h RO	**Rsvd1 (Rsvd1)**
16	0h RW	**Hardware LPM Enable (HLE)**

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RW	**Device Address (DA):** Note: This register is sticky.
7:4	0h RW	**Host Initiated Resume Duration (HIRD):** Note: This register is sticky.
3	0h RW	**Remote Wake Enable (RWE):** Note: This register is sticky.
2:0	0h RW	**L1 Status (L1S):** Note: This register is sticky.

16.82 Port X Hardware LPM Control Register (PORTHLPMC7)—Offset 4ECh

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters neccessary for xHC to automatically generate an LPM Token to the downstream device.

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**RESERVED (RSVD)**
13:10	0h RW	**Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1. The HIRDD value is is encoded as follows: 0h: 50 us (default) 1h: 125 us 2h: 200 us ... Fh: 1.175ms The value of 0h is interpreted as 50 us. Each incrementing value adds 75 us to the previous value.
9:2	0h RW	**L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer. This field shall be set to 00h by assertion of PR to '1'. Following are permissible values: 00h: 128 us (default) 01h: 256 us. ... FFh: 65,280us Note: This register is sticky.
1:0	0h RW	**Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used. Following are permissible values: 0: Initiate L1 using HIRD only time out (default) 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD 2,3: Reserved Note: This register is sticky.

16.83 Port Status and Control USB2 (PORTSC8)—Offset 4F0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.84 Port Power Management Status and Control USB2 (PORTPMSC8)—Offset 4F4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Port Test Control (PTC):** Note: This register is sticky.
27:17	0h RO	**Rsvd1 (Rsvd1)**
16	0h RW	**Hardware LPM Enable (HLE)**
15:8	0h RW	**Device Address (DA):** Note: This register is sticky.
7:4	0h RW	**Host Initiated Resume Duration (HIRD):** Note: This register is sticky.
3	0h RW	**Remote Wake Enable (RWE):** Note: This register is sticky.
2:0	0h RW	**L1 Status (L1S):** Note: This register is sticky.

16.85 Port X Hardware LPM Control Register (PORTHLPMC8)—Offset 4FCh

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters neccessary for xHC to automatically generate an LPM Token to the downstream device.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**RESERVED (RSVD)**
13:10	0h RW	**Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1. The HIRDD value is is encoded as follows: 0h: 50 us (default) 1h: 125 us 2h: 200 us ... Fh: 1.175ms The value of 0h is interpreted as 50 us. Each incrementing value adds 75 us to the previous value.
9:2	0h RW	**L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer. This field shall be set to 00h by assertion of PR to '1'. Following are permissible values: 00h: 128 us (default) 01h: 256 us. ... FFh: 65,280us Note: This register is sticky.
1:0	0h RW	**Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used. Following are permissible values: 0: Initiate L1 using HIRD only time out (default) 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD 2,3: Reserved Note: This register is sticky.

16.86 Port Status and Control USB2 (PORTSC9)—Offset 500h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.87 Port Power Management Status and Control USB2 (PORTPMSC9)—Offset 504h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Port Test Control (PTC):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
27:17	0h RO	**Rsvd1 (Rsvd1)**
16	0h RW	**Hardware LPM Enable (HLE)**
15:8	0h RW	**Device Address (DA):** Note: This register is sticky.
7:4	0h RW	**Host Initiated Resume Duration (HIRD):** Note: This register is sticky.
3	0h RW	**Remote Wake Enable (RWE):** Note: This register is sticky.
2:0	0h RW	**L1 Status (L1S):** Note: This register is sticky.

16.88 Port X Hardware LPM Control Register (PORTHLPMC9)—Offset 50Ch

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters neccessary for xHC to automatically generate an LPM Token to the downstream device.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**RESERVED (RSVD)**
13:10	0h RW	**Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1. The HIRDD value is is encoded as follows: 0h: 50 us (default) 1h: 125 us 2h: 200 us ... Fh: 1.175ms The value of 0h is interpreted as 50 us. Each incrementing value adds 75 us to the previous value.
9:2	0h RW	**L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer. This field shall be set to 00h by assertion of PR to '1'. Following are permissible values: 00h: 128 us (default) 01h: 256 us. ... FFh: 65,280us Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used. Following are permissible values: 0: Initiate L1 using HIRD only time out (default) 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD 2,3: Reserved Note: This register is sticky.

16.89 Port Status and Control USB3 (PORTSC10)—Offset 510h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.90 Port Power Management Status and Control USB3 (PORTPMSC10)—Offset 514h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RSVD)**
16	0h RW	**Force Link PM Accept (FLA)**
15:8	0h RW/S	**U2 Timeout (U2T)**
7:0	0h RW/S	**U1 Timeout (U1T)**

16.91 USB3 Port Link Info (PORTLI10)—Offset 518h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RO	**Link Error Count (LEC)**

16.92 Port Status and Control USB3 (PORTSC11)—Offset 520h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.93 Port Power Management Status and Control USB3 (PORTPMSC11)—Offset 524h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RSVD)**
16	0h RW	**Force Link PM Accept (FLA)**
15:8	0h RW/S	**U2 Timeout (U2T)**
7:0	0h RW/S	**U1 Timeout (U1T)**

16.94 USB3 Port Link Info (PORTLI11)—Offset 528h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RO	**Link Error Count (LEC)**

16.95 Port Status and Control USB3 (PORTSC12)—Offset 530h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.96 Port Power Management Status and Control USB3 (PORTPMSC12)—Offset 534h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RSVD)**
16	0h RW	**Force Link PM Accept (FLA)**
15:8	0h RW/S	**U2 Timeout (U2T)**
7:0	0h RW/S	**U1 Timeout (U1T)**

16.97 USB3 Port Link Info (PORTLI12)—Offset 538h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RO	**Link Error Count (LEC)**

16.98 Port Status and Control USB3 (PORTSC13)—Offset 540h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.99 Port Power Management Status and Control USB3 (PORTPMSC13)—Offset 544h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RSVD)**
16	0h RW	**Force Link PM Accept (FLA)**
15:8	0h RW/S	**U2 Timeout (U2T)**
7:0	0h RW/S	**U1 Timeout (U1T)**

16.100 USB3 Port Link Info (PORTLI13)—Offset 548h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RO	**Link Error Count (LEC)**

16.101 Port Status and Control USB3 (PORTSC14)—Offset 550h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.102 Port Power Management Status and Control USB3 (PORTPMSC14)—Offset 554h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RSVD)**
16	0h RW	**Force Link PM Accept (FLA)**
15:8	0h RW/S	**U2 Timeout (U2T)**
7:0	0h RW/S	**U1 Timeout (U1T)**

16.103 USB3 Port Link Info (PORTLI14)—Offset 558h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RO	**Link Error Count (LEC)**

16.104 Port Status and Control USB3 (PORTSC15)—Offset 560h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.105 Port Power Management Status and Control USB3 (PORTPMSC15)—Offset 564h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RSVD)**
16	0h RW	**Force Link PM Accept (FLA)**
15:8	0h RW/S	**U2 Timeout (U2T)**
7:0	0h RW/S	**U1 Timeout (U1T)**

16.106 USB3 Port Link Info (PORTLI15)—Offset 568h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RO	**Link Error Count (LEC)**

16.107 Port Status and Control USB3 (PORTSC16)—Offset 570h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.108 Port Power Management Status and Control USB3 (PORTPMSC16)—Offset 574h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RSVD)**
16	0h RW	**Force Link PM Accept (FLA)**
15:8	0h RW/S	**U2 Timeout (U2T)**
7:0	0h RW/S	**U1 Timeout (U1T)**

16.109 USB3 Port Link Info (PORTLI16)—Offset 578h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RO	**Link Error Count (LEC)**

16.110 Microframe Index (RTMFINDEX)—Offset 2000h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**Rsvd1 (Rsvd1)**
13:0	0h RO	**Microframe Index (IMAN0)**

16.111 Interrupter 1 Management (IMAN0)—Offset 2020h

There are 8 IMAN registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW	**Interrupt Enable (IE)**
0	0h RW/C	**Interrupt Pending (IP)**

16.112 Interrupter 1 Moderation (IMOD0)—Offset 2024h

There are 8 IMOD registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: FA0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Interrupt Moderation Counter (IMODC)**
15:0	FA0h RW	**Interrupt Moderation Interval (IMODI)**

16.113 Event Ring Segment Table Size 1 (ERSTSZ0)—Offset 2028h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RW	**Event Ring Segment Table Size (ERSTS)**

16.114 Event Ring Segment Table Base Address Low 1 (ERSTBA_LO0)—Offset 2030h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Event Ring Segment Table Base Address Register (ERSTBA)**
5:0	0h RO	**Rsvd1 (Rsvd1)**

16.115 Event Ring Segment Table Base Address High 1 (ERSTBA_HI0)—Offset 2034h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Segment Table Base Address (ERSTBA)**

16.116 Event Ring Dequeue Pointer Low 1 (ERDP_LO0)—Offset 2038h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RW	**Event Ring Dequeue Pointer (ERDP)**
3	0h RW/C	**Event Handler Busy (EHB)**
2:0	0h RW	**Dequeue ERST Segment Index (DESI)**

16.117 Event Ring Dequeue Pointer High 1 (ERDP_HI0)—Offset 203Ch

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Dequeue Pointer (ERDP)**

16.118 Interrupter 2 Management (IMAN1)—Offset 2040h

There are 8 IMAN registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW	**Interrupt Enable (IE)**
0	0h RW/C	**Interrupt Pending (IP)**

16.119 Interrupter 2 Moderation (IMOD1)—Offset 2044h

There are 8 IMOD registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: FA0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Interrupt Moderation Counter (IMODC)**
15:0	FA0h RW	**Interrupt Moderation Interval (IMODI)**

16.120 Event Ring Segment Table Size 2 (ERSTSZ1)—Offset 2048h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RW	**Event Ring Segment Table Size (ERSTS)**

16.121 Event Ring Segment Table Base Address Low 2 (ERSTBA_LO1)—Offset 2050h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Event Ring Segment Table Base Address Register (ERSTBA)**
5:0	0h RO	**Rsvd1 (Rsvd1)**

16.122 Event Ring Segment Table Base Address High 2 (ERSTBA_HI1)—Offset 2054h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Segment Table Base Address (ERSTBA)**

16.123 Event Ring Dequeue Pointer Low 2 (ERDP_LO1)—Offset 2058h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RW	**Event Ring Dequeue Pointer (ERDP)**
3	0h RW/C	**Event Handler Busy (EHB)**

Bit Range	Default & Access	Field Name (ID): Description
2:0	0h RW	**Dequeue ERST Segment Index (DESI)**

16.124 Event Ring Dequeue Pointer High 2 (ERDP_HI1)—Offset 205Ch

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Dequeue Pointer (ERDP)**

16.125 Interrupter 3 Management (IMAN2)—Offset 2060h

There are 8 IMAN registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW	**Interrupt Enable (IE)**
0	0h RW/C	**Interrupt Pending (IP)**

16.126 Interrupter 3 Moderation (IMOD2)—Offset 2064h

There are 8 IMOD registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: FA0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Interrupt Moderation Counter (IMODC)**
15:0	FA0h RW	**Interrupt Moderation Interval (IMODI)**

16.127 Event Ring Segment Table Size 3 (ERSTSZ2)—Offset 2068h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RW	**Event Ring Segment Table Size (ERSTS)**

16.128 Event Ring Segment Table Base Address Low 3 (ERSTBA_LO2)—Offset 2070h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Event Ring Segment Table Base Address Register (ERSTBA)**
5:0	0h RO	**Rsvd1 (Rsvd1)**

16.129 Event Ring Segment Table Base Address High 3 (ERSTBA_HI2)—Offset 2074h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Segment Table Base Address (ERSTBA)**

16.130 Event Ring Dequeue Pointer Low 3 (ERDP_LO2)—Offset 2078h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RW	**Event Ring Dequeue Pointer (ERDP)**
3	0h RW/C	**Event Handler Busy (EHB)**
2:0	0h RW	**Dequeue ERST Segment Index (DESI)**

16.131 Event Ring Dequeue Pointer High 3 (ERDP_HI2)—Offset 207Ch

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Dequeue Pointer (ERDP)**

16.132 Interrupter 4 Management (IMAN3)—Offset 2080h

There are 8 IMAN registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW	**Interrupt Enable (IE)**
0	0h RW/C	**Interrupt Pending (IP)**

16.133 Interrupter 4 Moderation (IMOD3)—Offset 2084h

There are 8 IMOD registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: FA0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Interrupt Moderation Counter (IMODC)**
15:0	FA0h RW	**Interrupt Moderation Interval (IMODI)**

16.134 Event Ring Segment Table Size 4 (ERSTSZ3)—Offset 2088h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RW	**Event Ring Segment Table Size (ERSTS)**

16.135 Event Ring Segment Table Base Address Low 4 (ERSTBA_LO3)—Offset 2090h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Event Ring Segment Table Base Address Register (ERSTBA)**
5:0	0h RO	**Rsvd1 (Rsvd1)**

16.136 Event Ring Segment Table Base Address High 4 (ERSTBA_HI3)—Offset 2094h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Segment Table Base Address (ERSTBA)**

16.137 Event Ring Dequeue Pointer Low 4 (ERDP_LO3)—Offset 2098h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RW	**Event Ring Dequeue Pointer (ERDP)**
3	0h RW/C	**Event Handler Busy (EHB)**

Bit Range	Default & Access	Field Name (ID): Description
2:0	0h RW	**Dequeue ERST Segment Index (DESI)**

16.138 Event Ring Dequeue Pointer High 4 (ERDP_HI3)—Offset 209Ch

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Dequeue Pointer (ERDP)**

16.139 Interrupter 5 Management (IMAN4)—Offset 20A0h

There are 8 IMAN registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW	**Interrupt Enable (IE)**
0	0h RW/C	**Interrupt Pending (IP)**

16.140 Interrupter 5 Moderation (IMOD4)—Offset 20A4h

There are 8 IMOD registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: FA0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Interrupt Moderation Counter (IMODC)**
15:0	FA0h RW	**Interrupt Moderation Interval (IMODI)**

16.141 Event Ring Segment Table Size 5 (ERSTSZ4)—Offset 20A8h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RW	**Event Ring Segment Table Size (ERSTS)**

16.142 Event Ring Segment Table Base Address Low 5 (ERSTBA_LO4)—Offset 20B0h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Event Ring Segment Table Base Address Register (ERSTBA)**
5:0	0h RO	**Rsvd1 (Rsvd1)**

16.143 Event Ring Segment Table Base Address High 5 (ERSTBA_HI4)—Offset 20B4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Segment Table Base Address (ERSTBA)**

16.144 Event Ring Dequeue Pointer Low 5 (ERDP_LO4)—Offset 20B8h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RW	**Event Ring Dequeue Pointer (ERDP)**
3	0h RW/C	**Event Handler Busy (EHB)**
2:0	0h RW	**Dequeue ERST Segment Index (DESI)**

16.145 Event Ring Dequeue Pointer High 5 (ERDP_HI4)—Offset 20BCh

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Dequeue Pointer (ERDP)**

16.146 Interrupter 6 Management (IMAN5)—Offset 20C0h

There are 8 IMAN registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW	**Interrupt Enable (IE)**
0	0h RW/C	**Interrupt Pending (IP)**

16.147 Interrupter 6 Moderation (IMOD5)—Offset 20C4h

There are 8 IMOD registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: FA0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Interrupt Moderation Counter (IMODC)**
15:0	FA0h RW	**Interrupt Moderation Interval (IMODI)**

16.148 Event Ring Segment Table Size 6 (ERSTSZ5)—Offset 20C8h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RW	**Event Ring Segment Table Size (ERSTS)**

16.149 Event Ring Segment Table Base Address Low 6 (ERSTBA_LO5)—Offset 20D0h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Event Ring Segment Table Base Address Register (ERSTBA)**
5:0	0h RO	**Rsvd1 (Rsvd1)**

16.150 Event Ring Segment Table Base Address High 6 (ERSTBA_HI5)—Offset 20D4h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Segment Table Base Address (ERSTBA)**

16.151 Event Ring Dequeue Pointer Low 6 (ERDP_LO5)—Offset 20D8h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RW	**Event Ring Dequeue Pointer (ERDP)**
3	0h RW/C	**Event Handler Busy (EHB)**

Bit Range	Default & Access	Field Name (ID): Description
2:0	0h RW	**Dequeue ERST Segment Index (DESI)**

16.152 Event Ring Dequeue Pointer High 6 (ERDP_HI5)—Offset 20DCh

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Dequeue Pointer (ERDP)**

16.153 Interrupter 7 Management (IMAN6)—Offset 20E0h

There are 8 IMAN registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW	**Interrupt Enable (IE)**
0	0h RW/C	**Interrupt Pending (IP)**

16.154 Interrupter 7 Moderation (IMOD6)—Offset 20E4h

There are 8 IMOD registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: FA0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Interrupt Moderation Counter (IMODC)**
15:0	FA0h RW	**Interrupt Moderation Interval (IMODI)**

16.155 Event Ring Segment Table Size 7 (ERSTSZ6)—Offset 20E8h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RW	**Event Ring Segment Table Size (ERSTS)**

16.156 Event Ring Segment Table Base Address Low 7 (ERSTBA_LO6)—Offset 20F0h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Event Ring Segment Table Base Address Register (ERSTBA)**
5:0	0h RO	**Rsvd1 (Rsvd1)**

16.157 Event Ring Segment Table Base Address High 7 (ERSTBA_HI6)—Offset 20F4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Segment Table Base Address (ERSTBA)**

16.158 Event Ring Dequeue Pointer Low 7 (ERDP_LO6)—Offset 20F8h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RW	**Event Ring Dequeue Pointer (ERDP)**
3	0h RW/C	**Event Handler Busy (EHB)**
2:0	0h RW	**Dequeue ERST Segment Index (DESI)**

16.159 Event Ring Dequeue Pointer High 7 (ERDP_HI6)—Offset 20FCh

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Dequeue Pointer (ERDP)**

16.160 Interrupter 8 Management (IMAN7)—Offset 2100h

There are 8 IMAN registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW	**Interrupt Enable (IE)**
0	0h RW/C	**Interrupt Pending (IP)**

16.161 Interrupter 8 Moderation (IMOD7)—Offset 2104h

There are 8 IMOD registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: FA0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Interrupt Moderation Counter (IMODC)**
15:0	FA0h RW	**Interrupt Moderation Interval (IMODI)**

16.162 Event Ring Segment Table Size 8 (ERSTSZ7)—Offset 2108h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RW	**Event Ring Segment Table Size (ERSTS)**

16.163 Event Ring Segment Table Base Address Low 8 (ERSTBA_LO7)—Offset 2110h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Event Ring Segment Table Base Address Register (ERSTBA)**
5:0	0h RO	**Rsvd1 (Rsvd1)**

16.164 Event Ring Segment Table Base Address High 8 (ERSTBA_HI7)—Offset 2114h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Segment Table Base Address (ERSTBA)**

16.165 Event Ring Dequeue Pointer Low 8 (ERDP_LO7)—Offset 2118h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RW	**Event Ring Dequeue Pointer (ERDP)**
3	0h RW/C	**Event Handler Busy (EHB)**

Bit Range	Default & Access	Field Name (ID): Description
2:0	0h RW	**Dequeue ERST Segment Index (DESI)**

16.166 Event Ring Dequeue Pointer High 8 (ERDP_HI7)—Offset 211Ch

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Dequeue Pointer (ERDP)**

16.167 Door Bell 1 (DB0)—Offset 3000h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.168 Door Bell 2 (DB1)—Offset 3004h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.169 Door Bell 3 (DB2)—Offset 3008h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.170 Door Bell 4 (DB3)—Offset 300Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.171 Door Bell 5 (DB4)—Offset 3010h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.172 Door Bell 6 (DB5)—Offset 3014h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.173 Door Bell 7 (DB6)—Offset 3018h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.174 Door Bell 8 (DB7)—Offset 301Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.175 Door Bell 9 (DB8)—Offset 3020h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.176 Door Bell 10 (DB9)—Offset 3024h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.177 Door Bell 11 (DB10)—Offset 3028h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.178 Door Bell 12 (DB11)—Offset 302Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.179 Door Bell 13 (DB12)—Offset 3030h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.180 Door Bell 14 (DB13)—Offset 3034h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.181 Door Bell 15 (DB14)—Offset 3038h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.182 Door Bell 16 (DB15)—Offset 303Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.183 Door Bell 17 (DB16)—Offset 3040h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.184 Door Bell 18 (DB17)—Offset 3044h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.185 Door Bell 19 (DB18)—Offset 3048h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.186 Door Bell 20 (DB19)—Offset 304Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.187 Door Bell 21 (DB20)—Offset 3050h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.188 Door Bell 22 (DB21)—Offset 3054h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.189 Door Bell 23 (DB22)—Offset 3058h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.190 Door Bell 24 (DB23)—Offset 305Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.191 Door Bell 25 (DB24)—Offset 3060h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.192 Door Bell 26 (DB25)—Offset 3064h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.193 Door Bell 27 (DB26)—Offset 3068h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.194 Door Bell 28 (DB27)—Offset 306Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.195 Door Bell 29 (DB28)—Offset 3070h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.196 Door Bell 30 (DB29)—Offset 3074h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.197 Door Bell 31 (DB30)—Offset 3078h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.198 Door Bell 32 (DB31)—Offset 307Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.199 Door Bell 32 (DB32)—Offset 3080h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.200 XECP_SUPP_USB2_0 (XECP_SUPP_USB2_0)—Offset 8000h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 2000802h

Bit Range	Default & Access	Field Name (ID): Description
31:24	2h RO	**USB Major Revision: 2.0 (USB2_MAJ_REV)**
23:16	0h RO	**USB Minor Revision (USB_MIN_REV)**
15:8	8h RO	**Next Capability Pointer (NCP)**
7:0	2h RO	**Supported Protocol ID (SPID)**

16.201 XECP_SUPP_USB2_1 (XECP_SUPP_USB2_1)—Offset 8004h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 20425355h

Bit Range	Default & Access	Field Name (ID): Description
31:0	20425355h RO	**XECP_SUPP_USB2_1 (XECP_SUPP_USB2_1):** Namestring USB

16.202 XECP_SUPP_USB2_2 (XECP_SUPP_USB2_2)—Offset 8008h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 30190901h

Bit Range	Default & Access	Field Name (ID): Description
31:28	3h RO	**Protocol Speed ID Count (PROT_SPD_ID_CNT):** 3 USB 2.0 Speed (High, Full, Low)
27:21	0h RO	**Rsvd0 (Rsvd0)**
20	1h RW/L	**BESL LPM Capability (BLC):** Bit is set to 1 to indicate that the the ports described by this xHCI Supported Protocol Capability will apply BESL timing to BESL and BESLD fields of the PORTPMSC and PORTHLPMCC registers.
19	1h RW/L	**Protocol Defined - Hardware LMP Capability (HLC)**
18	0h RO	**Protocol Defined - Integrated Hub Implementation (IHI)**

Bit Range	Default & Access	Field Name (ID): Description
17	0h RO	**Protocol Defined - High SPeed Only (HSO)**
16	1h RO	**Reserved (RSVD)**
15:8	9h RO	**Compatible Port Count (CPC)**
7:0	1h RO	**Compatible Port Offset (CPO)**

16.203 XECP_SUPP_USB2_3 (Full Speed) (XECP_SUPP_USB2_3)—Offset 8010h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: C0021h

Bit Range	Default & Access	Field Name (ID): Description
31:16	Ch RO	**Protocol Speed ID Mantissa (PSIM)**
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	0h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	2h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	1h RO	**Protocol Speed ID Value (PSIV)**

16.204 XECP_SUPP_USB2_4 (Low Speed) (XECP_SUPP_USB2_4)—Offset 8014h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 5DC0012h

Bit Range	Default & Access	Field Name (ID): Description
31:16	5DCh RO	**Protocol Speed ID Mantissa (PSIM)**

Bit Range	Default & Access	Field Name (ID): Description
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	0h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	1h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	2h RO	**Protocol Speed ID Value (PSIV)**

16.205 XECP_SUPP_USB2_5 (High Speed) (XECP_SUPP_USB2_5)—Offset 8018h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 1E00023h

Bit Range	Default & Access	Field Name (ID): Description
31:16	1E0h RO	**Protocol Speed ID Mantissa (PSIM)**
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	0h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	2h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	3h RO	**Protocol Speed ID Value (PSIV)**

16.206 XECP_SUPP_USB3_0 (XECP_SUPP_USB3_0)—Offset 8020h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 3001402h

Bit Range	Default & Access	Field Name (ID): Description
31:24	3h RO	**USB Major Revision: 3.0 (USB3_MAJ_REV)**
23:16	0h RO	**USB Minor Revision (USB_MIN_REV)**
15:8	14h RO	**Next Capability Pointer (NCP)**
7:0	2h RO	**Supported Protocol ID (SPID)**

16.207 XECP_SUPP_USB3_1 (XECP_SUPP_USB3_1)—Offset 8024h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 20425355h

Bit Range	Default & Access	Field Name (ID): Description
31:0	20425355h RO	**XECP_SUPP_USB2_1 (XECP_SUPP_USB2_1):** Namestring USB

16.208 XECP_SUPP_USB3_2 (XECP_SUPP_USB3_2)—Offset 8028h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 7000070Ah

Bit Range	Default & Access	Field Name (ID): Description
31:28	7h RO	**Protocol Speed ID Count (PROT_SPD_ID_CNT):** 1 USB 3.0 Speed (Supper Speed)
27:16	0h RO	**Rsvd0 (Rsvd0)**
15:8	7h RO	**Compatible Port Count (CPC)**
7:0	Ah RO	**Compatible Port Offset (CPO)**

16.209 XECP_SUPP_USB3_3 (XECP_SUPP_USB3_3)—Offset 8030h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 4E00121h

Bit Range	Default & Access	Field Name (ID): Description
31:16	4E0h RO	**Protocol Speed ID Mantissa (PSIM)**
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	1h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	2h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	1h RO	**Protocol Speed ID Value (PSIV)**

16.210 XECP_SUPP_USB3_4 (XECP_SUPP_USB3_4)—Offset 8034h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 9C00122h

Bit Range	Default & Access	Field Name (ID): Description
31:16	9C0h RO	**Protocol Speed ID Mantissa (PSIM)**
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	1h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	2h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	2h RO	**Protocol Speed ID Value (PSIV)**

16.211 XECP_SUPP_USB3_5 (XECP_SUPP_USB3_5)—Offset 8038h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 13800123h

Bit Range	Default & Access	Field Name (ID): Description
31:16	1380h RO	**Protocol Speed ID Mantissa (PSIM)**
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	1h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	2h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	3h RO	**Protocol Speed ID Value (PSIV)**

16.212 XECP_SUPP_USB3_6 (XECP_SUPP_USB3_6)—Offset 803Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 50134h

Bit Range	Default & Access	Field Name (ID): Description
31:16	5h RO	**Protocol Speed ID Mantissa (PSIM)**
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	1h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	3h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	4h RO	**Protocol Speed ID Value (PSIV)**

16.213 XECP_SUPP_USB3_7 (XECP_SUPP_USB3_7)—Offset 8040h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 5B10125h

Bit Range	Default & Access	Field Name (ID): Description
31:16	5B1h RO	**Protocol Speed ID Mantissa (PSIM)**
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	1h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	2h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	5h RO	**Protocol Speed ID Value (PSIV)**

16.214 XECP_SUPP_USB3_8 (XECP_SUPP_USB3_8)—Offset 8044h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: B630126h

Bit Range	Default & Access	Field Name (ID): Description
31:16	B63h RO	**Protocol Speed ID Mantissa (PSIM)**
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	1h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	2h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	6h RO	**Protocol Speed ID Value (PSIV)**

16.215 XECP_SUPP_USB3_9 (XECP_SUPP_USB3_9)—Offset 8048h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 16C60127h

Bit Range	Default & Access	Field Name (ID): Description
31:16	16C6h RO	**Protocol Speed ID Mantissa (PSIM)**
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	1h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	2h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	7h RO	**Protocol Speed ID Value (PSIV)**

16.216 Host Controller Capability (HOST_CTRL_CAP_REG)—Offset 8070h

This is a register that describe the host controller the extended cap location. It includes the , XECP_HOST_NEXT_CAP_OFFSET and VEND_DEF_HOST_CAP_ID_192. This register is not subject to HW save and restore.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 4DFFC0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD)**
23:16	4Dh RW/L	**Valid Length (VALID_LENGTH):** Indicates the number of valid DWords in the capability, that need to be saved and restored. 47h -> 818Bh is the last valid byte
15:8	FFh RW/L	**Next Capability Pointer (XECP_HOST_NEXT_CAP_OFFSET)**
7:0	C0h RW/L	**Supported Protocol ID (VEND_DEF_HOST_CAP_ID)**

16.217 Override EP Flow Control (HOST_CLR_MASK_REG)—Offset 8078h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RO	**RESERVED (RSVD)**
9:5	0h WO	**Slot Number Default Config (SNDC):** 5bits of slot number as a default configuration. It can scale to max of 128 slots
4:1	0h WO	**EP Number (EP_NUM):** 4bits of EP number
0	0h WO	**Clear Internal Scheduler's Mask (CISM):** This is a register that is used to clear the internal scheduler's mask that is used to stop scheduling a particular EP. Bit0 indicates the direction of the EP

16.218 Clear Active IN EP ID Control (HOST_CLR_IN_EP_VALID_REG)—Offset 807Ch

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h WO	**Clear Active IN EP ID Control (HOST_CLR_IN_EP_VALID_REG):** This register is used to clear the internal valid IN EP array that TRM stored in order to guarantee one IN EP per port. This register allows software to clear the valid bit of each port IN EP. This field indicates the port number. For a 2port configuration, only bit1:0 are valid. It can scale for the max number of ports that we support.

16.219 Clear Poll Mask Control (HOST_CLR_PMASK_REG)—Offset 8080h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RO	**RESERVED (RSVD)**
9:5	0h WO	**Slot Number Default Config (SNDC):** 5bits of slot number as a default configuration. It can scale to max of 128 slots
4:1	0h WO	**EP Number (EP_NUM):** 4bits of EP number
0	0h WO	**Clear Internal Scheduler's Poll Mask (CISPM):** This is a register that is used to clear the internal scheduler's poll mask that is used to indicate whether we need to poll this EP. This is used for USB2. Bit0 indicates the direction of the EP

16.220 Host Control Scheduler (HOST_CTRL_SCH_REG)—Offset 8094h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 8100h

Bit Range	Default & Access	Field Name (ID): Description
31:21	0h RW	**Scheduler Host Control Reg Cont (SCHED_HOST_CTRL_CONT):** 31:25 Reserved (24): disable marking overlap flag on all TT periodic INs. (23): disable async. scheduling while periodic active to same port (22): disable "level" method of USB2 port periodic done check (on by default) (21): enable "strobe" method of USB2 port periodic done check (off by default)
20:13	4h RW	**TTE Host Control (TTE_HOST_CTRL):** (0): disable interrupt complete split limit to 3 microframes (1): disable checking of missed microframes (2): disable split error request w/NULL pointer on speculative INs with data payload and no TRB. (3): disable deferred split error request on speculative IN with data payload and no TRB. (7:4): reserved
12:11	0h RW	**Cache Size Control Reg (CACHE_SZ_CTRL):** 0: 64 1: 32 2,3: 16
10:9	0h RW	**Maximum EP Per Slot (MAX_EP_SLOT):** 0: 32 1: 16 2: 8 3: 4
8	1h RW	**Turn on scratch_pad_en (TO_SCRATCH_PAD_EN)**

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Scheduler Host Control Reg (SCHED_HOST_CTRL):** (0): disable poll delay (1): disable TRM active in EP valid check (2): enable TTE overlap prevention on interrupt IN EPs (at cost of possible service interval slip) (3) enable TTE overlap prevention on interrupt OUT EPs (at cost of possible service interval slip) (5:4) scheduler sort pattern 00 (default) search ISO ahead of interrupt within each service interval 01 - search USB2-ISO, USB3-ISO, USB2-Interrupt, USB3-Interrupt within each service interval 10 - search strictly by interval 11 - search all ISO intervals ahead interrupt intervals and within each interval, USB2 ahead of USB3 (6): disable 1 pack scheduling limit when ISO pending in present microframe (7): enable check to stop scheduling on port that are not connected

16.221 Global Port Control (HOST_CTRL_PORT_CTRL)—Offset 80A0h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 380Fh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**OVERFLOW ERROR DETECTION ENABLE (OVERFLOW_ERR_DETECT_EN)**
30:12	3h RW	**HBUF_WATER_MARK (HBUF_WATER_MARK)**
11	1h RW	**CPL Cut Thru Enable (CPL_CUT_THRU_EN)**
10:4	0h RW	**RESERVED (RSVD0)**
3:0	Fh RW	**RESERVED (RSVD1)**

16.222 Power Management Control (PMCTRL_REG)—Offset 80A4h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 2DFF90h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Async PME Source Enable (ASYNC_PME_SRC_EN):** This field allows the async PME source to be allowed to generate PME. This is specifically required for SOCs that do not allow for any clock other than RTC to be available during RTD3.
30	0h RW	**Legacy PME Source Enable (LEGACY_PME_SRC_EN):** This field allows the legacy PME source to be used in PME generation. The legacy source in in reference to the source prior to the RTD3 changes.
29	0h RW	**Reset Warn Power Gate Trigger Disable (RESET_WARN_PWR_GATE_TRIGGER_DISABLE):** This field controls the actions taken for due to reset warn. 0 - Reset Warn will trigger a HW autonomous Power Gate 1 - Reset Warn will not trigger a HW autonomous Power Gate
28	0h RW	**CLR_PME_FLAG_PULSE_AUX_CCLK (CLR_PME_FLAG_PULSE_AUX_CCLK)**
27	0h RW	**RESERVED (RSVD)**
26	0h RW	**XLFPSCOUNTSRC (XLFPSCOUNTSRC):** XLFPSCOUNTSRC (Source for LFPS OFF Counter) 0: Central RTC Counter for LFPS detection 1: Local Counter for LFPS detection
25	0h RW	**XELFPSRTC (XELFPSRTC):** XELFPSRTC (Enable LFPS Filtering on RTC) 0: Use Oscillator clock for LFPS Filtering during P3 1: Use RTC Clock for LFPS Filtering during P3
24	0h RW	**XMPHYSPGDD0I2 (XMPHYSPGDD0I2):** XMPHYSPGDD0I2 (ModPhy Sus Well Power Gate Disable for D0I2) 0 : Modphy sus well power gating enabled 1 : Modphy sus well power gating disabled
23	0h RW	**XMPHYSPGDD0I3 (XMPHYSPGDD0I3):** XMPHYSPGDD0I3 (ModPhy Sus Well Power Gate Disable for D0I3) 0 : Modphy sus well power gating enabled 1 : Modphy sus well power gating disabled
22	0h RW	**XMPHYSPGDRTD3 (XMPHYSPGDRTD3):** XMPHYSPGDRTD3 (ModPhy Sus Well Power Gate Disable for RTD3) 0 : Modphy sus well power gating enabled 1 : Modphy sus well power gating disabled
21:18	Bh RW	**XD3RTCPTTM (XD3RTCPTTM):** XD3RTCPTTM (D3 RTC Port Timer Tick Multiplier) This register will be the multiplication factor for determining SSIC Wake Detection Frequency and RXDET based on the XD3RTCPTTC value. If XD3RTCPTTC is 9h and this register is Bh, frequency for RXDET and SSIC H8EXIT detection while MODPHY SUS Power gating is enabled would be 99ms.
17	0h RW	**U3 LFPS Periodic Sampling ON Time Control (U3_LFPS_PRDC_SAMPLING_ON_TIME_CTRL):** This field controls the ON time for the LFPS periodic sampling for USB3/SSIC ports. 0 ON time is 2 rtc clocks 1 ON time is 3 rtc clocks Note: This field is ignored if USB3/SSIC PHY SUS Well Power Gating is enabled.

Bit Range	Default & Access	Field Name (ID): Description
16	1h RW	**AON LFPS Detector Enable Mode (AON_LFPS_DETECTOR_EN_MODE):** 1 - Allow the LFSP Detector in AON to own LFPS detection when the port is in PS3 for U2/U3 - not RxD regardless of port ownership. 0 - Allow the LFPS Detector in AON to own the LFPS detection only when the AON owns the port and in U2/U3 - not RxD
15:8	FFh RW	**SS U3 LFPS Detection Threshold (SS_U3_LFPS_DETECTION_THRESHOLD):** This field controls the threshold used to determine when a valid U3 Wake is detected through when using the unfiltered LFPS source. The value on this field will reflect the binary count required to have been detected on the counter being clocked by the unfiltered lfps source to result in a valid U3 wake detection.
7:4	9h RW	**SS_U3_LFPS_PRDC_SAMPLING_OFFTIME_CTRL (SS_U3_LFPS_PRDC_SAMPLING_OFFTIME_CTRL):** This field controls the OFF time for the LFPS periodic sampling for USB3 Ports 0x0 periodic sampling is disabled. 0x1 OFF time is 1ms 0x2 OFF time is 2ms 0xF OFF time is 15ms The ON Time is determined by the amount of time required to reliably determine if there is a valid LFPS and is HW implementation specific. A speed up mode shall be implemented where this field is in units of us. i.e. 0x1 = 1 us OFF time, 0x2 = 2 us OFF time, etc.
3	0h RW	**PS3 LFPS Source Select (PS3_LFPS_SRC_SEL):** 0 LFPS Source is unfiltered 1 LFPS Source is filtered (Rx-Elec-Idle) LFPS Source is Rx-Elec-Idle for any non PS3 state.
2	0h RW	**XHCI Engine Autonomous Power Gate Exit Reset Policy (XHC_AUTO_PWRGATE_EXITRST_POLICY):** Controls when the xHCI engine is brought out of reset due to a power ungate. 0 Engine is brought out of reset when D3 to D0 is triggered. This allows for a quick power up sequence while leaving the virtual PCIe LTSSM in L23 is power ungate is not due to D3 to D0. 1 Engine is brought out of reset along with the rest of the IP. This is required for PMC save/restore flow.
1	0h RW	**USB2 Port Wake Unit Coupling Policy (USB2_PORT_WAKE_COUPLING_POLICY):** Controls the trigger for USB2 Port Wake Units to initiate Port Level Power Off Preparation. 0 RTD3 triggered 1 - Port Triggered when in L1, L2 or Disabled, Disconnected
0	0h RW	**USB3 Port Wake Unit Coupling Policy (USB3_PORT_WAKE_COUPLING_POLICY):** Controls the trigger for USB3 Port Wake Units to initiate Port Level Power Off Preparation. 0 - RTD3 Triggered 1 - Port Triggered when in PS3 due to RxDetect, U3, U2 or Disabled

16.223 PGCB Control (PGCBCTRL_REG)—Offset 80A8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 315555h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**RESERVED (RSVD)**
27:25	0h RW	**IP_INACCESSIBLE_HYSTERESIS_TIMER (IP_INACCESSIBLE_HYSTERESIS_TIMER):** 000 ? 50us 001 ? 100us 010 ? 150us 011 ? 200us 000 ? 250us 001 ? 300us 010 ? 350us 111 ? 400us
24	0h RW	**Override_Disable (OVERRIDE_DISABLE)**
23:20	3h RW	**CDH Aggregation Minimum Wait Time (CDH_MIN_WAIT_TIME):** This field controls the minimum time that we must wait for resets to propagate before allowing Power Up flow to complete. The Power Up flow requires special handling of clocks, reset and sleep signaling which requires the CDH to monitor reset propagation. This timer allow for flexibility on when we consider all resets propagated. 0h Disabled 1h 2 clocks 2h 4 clocks 3h 8 clocks 4h 16 clocks 5h 32 clocks 6h 64 clocks 7h 128 clocks
19	0h RW	**RESERVED (CDH_RST_PROPAGATION_CNTRL):** This bit will indicate the mode of operation for the CDH Reset Propagation Aggregator. It allows for a mode where the aggregation will wait for an indication from each CDH before proceeding OR ignore the CDH indication before proceeding. Regardless of the mode, the aggregator will enforce the CDH Aggregation Mini Wait Time. 0 Aggregate all the CDH indications before triggering the CDH Aggregation Mini Wait Time. Once the Min Wait Time is met a notification to the Power Sequencer/Control will be initiated 1 Trigger the CDH Aggregation Mini Wait Time w/o waiting for CDH indication. Once the Min Wait time is met a notification to the Power Sequencer/Control will be initiated.
18	0h RW	**MMP_PFET_REQ_OVRD (MMP_PFET_REQ_OVRD):** This bit will disable the MMP PFET request condition for PGCB control. 1 MMP PFET Request Ignored 0 Default
17:16	1h RW	**PGCB Clock Gate Request to PGCB Sleep (PGCB_CLK_GATE_REQ_2_PGCB_SLEEP):** Value representing the minimum number of delay clocks required between the assertion of pgcb_ip_clkgate_b to the deassertion of pgcb_sleep on PG exit. (This is only applicable for IP-Accessible PG with state-retention enabled.) Please see the description of cfg_tsleepact for more details.

Bit Range	Default & Access	Field Name (ID): Description
15:14	1h RW	**PGCB Sleep Deassertion to PGCB ISM Unlock Req (PGCB_SLEEP_DEASRTN_2_ISM_UNLOCK_REQ):** For IP-Accessible PG (with state retention): Value representing the minimum number of delay clocks required between the deassertion of gcb_sleepto the deassertion of pgcb_ip_*_lock_req_b. For IP-Inaccessible PG (or with state-retention disabled): Value representing the minimum number of delay clocks required between the deassertion of pgcb_sleep to the deassertion of pgcb_isol_en_b.
13:12	1h RW	**PGCB Prim Reset Deassertion to PGCB Next State (PGCB_PRIMRST_DEASRTN_2_NSTATE):** For IP-Accessible PG (with state retention): Value representing the minimum number of delay clocks required between the deassertion of pgcb_sleep to the deassertion of pgcb_ip_*_lock_req_b. For IP-Inaccessible PG (or with state-retention disabled): Value representing the minimum number of delay clocks required between the deassertion of pgcb_sleep to the deassertion of pgcb_isol_en_b.
11:10	1h RW	**PGCB Side Reset Deassertion to PGCB Prim Reset Deassertion (PGCB_SIDERST_DEASRTN_2_PRIMERST_DEASRTN):** Value representing the number of delay clocks required between the deassertion of gcb_force_rst_bto the deassertion of gcb_force_prim_rst_b (This is only applicable for IP-Accessible PG.) Please see the description of cfg_tsleepact for more details.
9:8	1h RW	**PGCB Latch Isolation High to PGCB Clock Ungate Request (PGCB_LATCH_ISO_HI_2_PGCB_CLK_UNGATE_REQ):** Value representing the number of delay clocks required between the assertion of pgcb_isol_latchen to the deassertion of pgcb_ip_clkgate_req_b. Please see the description of cfg_tsleepact for more details.
7:6	1h RW	**PGCB Isolation Deassertion to PGCB Latch Isolation High Count (PGCB_ISO_DEASRTN_2_PGCB_ISO_HI_CNT):** Value representing the minimum number of delay clocks required between the deassertion of pgcb_isol_en_b to the assertion of gcb_isol_latchen. Please see the description of cfg_tsleepact for more details.
5:4	1h RW	**PGCB Reset Assertion to PGCB Power Down Request Count (PGCB_RST_2_PGCB_PWRDOWN_REQCNT):** For IP-Accessible PG (with state retention): Value representing the minimum number of delay clocks required between the assertion of pgcb_force_prim_rst_b and pgcb_force_rst_b to the assertion of pgcb_pmc_pg_req_b. For IP-Inaccessible PG (or with state-retention disabled): Value representing the minimum number of delay clocks required between the assertion of pgcb_force_prim_rst_b and pgcb_force_rst_b to the assertion of pgcb_sleep. Please see the description of cfg_tsleepact for more details.

Bit Range	Default & Access	Field Name (ID): Description
3:2	1h RW	**PGCB Isolation Assertion to PGCB Reset Assertion Count (PGCB_ISO_ASRTN_2_PGCB_RST_ASRTN_CNT):** Value representing the minimum number of delay clocks required between the assertion of gcb_isol_en_bto the assertion of gcb_force_prim_rst_band gcb_force_rst_b Please see the description of [1:0] for more details.
1:0	1h RW	**PGCB Sleep Assertion to PGCB Isolation Enable Count (PGCB_SLEEP_ASRTN_2_PGCB_ISO_EN_CNT):** For IP-Accessible PG (with state retention): Value representing the minimum number of delay clocks required between the assertion of pgcb_sleep (and the deassertion of pgcb_isol_latchen) to the assertion of pgcb_isol_en_b. For IP-Inaccessible PG (or with state-retention disabled): Value representing the minimum number of delay clocks required between the deassertion of pgcb_isol_latchen to the assertion of pgcb_isol_en_b, as well as the minimum number of delay clocks required between the assertion of pgcb_sleep to the assertion of pgcb_pmc_pg_req_b. Common for all cfg_t* inputs: 0 1 clock 1 2 clocks 0 8 clocks 1 256 clocks For any of the cfg_t* inputs that an IP may feel are a don-care and the IP does not feel they will be useful for post-silicon debug, the recommendation is to tie the input to 201 (2 clocks). Otherwise, the inputs may be tied to another relevant value or connected to a configuration register. Note: These signals may only change while pgcb_pwrupidle is asserted, if pgcb_pwrupidle is deasserted it should be valid and stable. (It may change the same cycle that ip_pgcb_pg_rdy_req_b asserts.)

16.224 D0I3 Control (DOI3CTRL_REG)—Offset 80ACh

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 8h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**RESERVED (RSVD)**
3	1h RW/1C	**RestoreRequired (RESTORE_REQUIRED):** When set (by HW), SW must restore state to the IP. The state may have been lost due to a reset or full power lost. SW clears the bit by writing a 1. This bit will be set on initial power up.
2	0h RW	**D0I3 (D0I3):** SW sets this bit to 1 to move the IP into the D0i3 state. Writing this bit to 0 will return the IP to the fully active D0 state (D0i0).
1:0	0h RO	**RESERVED (RSVD1)**

16.225 HOST_CTRL_MISC_REG (HOST_CTRL_MISC_REG)—Offset 80B0h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 1037Fh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**USB2_LTRUPDT_DIS (USB2_LTRUPDT_DIS)**
30	0h RW	**USB2 Line State Debounce During Port Reset Policy (USB2_LINE_STATE_DEBOUNCE_DURING_PORT_RESET_POLICY):** This register controls how the debounce is enforced during the Port Reset phase. 0 do not enable the line state debounce during port reset. 1 enable the line state debounce during port reset.
29	0h RW	**TTE PEXE Credit Fix Disable (TTE_PEXE_CREDIT_FIX_DISABLE):** When set, it disables a fix implemented to re-deem PEXE credits when a port is disconnected
28	0h RW	**TTE Scheduling policy (TTE_SCHEDULING_POLICY):** This register controls a fix made to prevent over-scheduling by not account for 188B in each uFrame. Setting this bit will disable the fix and allow for over-scheduling.
27	0h RW	**USB3 ITP Delta Timer Source Select (USB3_ITP_DELTA_TIMER_SOURCE_SELECT):** This register selects the source for the delta timer tracking used for ITP generation. 0 the source is a 16.666 ns tick generated from a crystal reference clock 1 - the source is a 16.666 ns tick generated from the aux_cclk. This field needs to remain in sync with Frame Timer Source Select to ensure the are both set or both cleared. There is no support for any other combination.
26	0h RW	**Frame Timer Source Select (FRAME_TIMER_SOURCE_SELECT):** This register controls the source for the frame timer. 0 the source for the frame timer is a crystal reference clock 1 the source for the frame timer is the aux_cclk.
25	0h RW	**uFrame Masking Enable (UFRAME_MASKING_ENABLE):** If set, enables the uFrame tick to be masked due to ports being in U3/NC. This controls a fix made to disable the auto masking of uFrame tick due to port state w/o any pipeline idle condition. When cleared, we rely on gating of frame timer due to proper port state and idleness tracking from the pipeline.
24	0h RW	**Late FID Check Disable (LATE_FID_CHECK_DISABLE):** This register disables the Late FID Check performed when starting an ISOCH stream.

Bit Range	Default & Access	Field Name (ID): Description
23:20	0h RW	**RESERVED (RSVD1)**
19	0h RW	**USB2 Resume Cx Inhibit Disable (USB2_RESUME_CX_INHIBIT_DISABLE):** Controls if USB2 L1 Resume is allowed to contribute to DMA Active which will inhibit Cx state. 0 USB2 L1 Resume is allowed to inhibit Cx via DMA Active 1 USB2 L1 Resume is NOT allowed to inhibit Cx via DMA Active When cleared, Cx will only be inhibited when the DMA traffic for the port begins.
18:16	1h RW	**Extra uFrame (EXTRA_UFRAME):** This register controls the extra number of uFrames added onto the advancing of late FID check.
15:0	37Fh RW	**Valid Isoch Scheduling Range (VALID_ISOCH_SCHEDULING_RANGE):** This register defines the window in miliseconds from the current Frame that will be considered for scheduling in an upcoming Frame. Anything scheduled outside of this window will be considered as late and will trigger the Missed Service Error.

16.226 HOST_CTRL_MISC_REG2 (HOST_CTRL_MISC_REG2)—Offset 80B4h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RW	**max_short_pkt_adv_cnt (MAX_SHORT_PKT_ADV_CNT)**
28	0h RW	**dis_sch_frameid_chk (DIS_SCH_FRAMEID_CHK)**
27:13	0h RW	**misc_config_reg2_27_13_rsvd (RSVD1)**
12	0h RW	**disable_idt_fix_odma (DISABLE_IDT_FIX_ODMA):** arc fix 0:enable the fix 1:disable the fix
11	0h RW	**disable_ping_fix_odma (DISABLE_PING_FIX_ODMA):** fix 0:enable the fix 1:disable the fix
10	0h RW	**disable_cerr_fix_idma 0: fix is enabled 1:fix is disabled (DISABLE_CERR_FIX_IDMA)**
9	0h RW	**en_100ms_watch_dog_timer (EN_100MS_WATCH_DOG_TIMER)**
8	0h RW	**en_watch_dog_timer (EN_WATCH_DOG_TIMER)**

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**misc_config_reg2_bit7_rsvd (RSVD2)**
6	0h RW	**disable_tcg_ungate_on_flush (DISABLE_TCG_UNGATE_ON_FLUSH)**
5	0h RW	**disable_vnn_frame_timer (DISABLE_VNN_FRAME_TIMER)**
4	0h RW	**disable_clr_ccs_on_cas_set (DISABLE_CLR_CCS_ON_CAS_SET)**
3	0h RW	**disable_rhub_park_at_dbcDisc (DISABLE_RHUB_PARK_AT_DBCDISC)**
2	0h RW	**disable_block_wpr_on_disPorts (DISABLE_BLOCK_WPR_ON_DISPORTS)**
1:0	0h RW	**HOST_CTRL_MISC_REG2_bits1_0 (HOST_CTRL_MISC_REG2_1_0)**

16.227 SSPE_REG (SSPE_REG)—Offset 80B8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**ssCfgBlockPwrDwn4ActLFPS (SS_CFG_BLOCK_PWRDWN_4_ACT_LFPS)**
30	0h RW	**dis_clr_ccs_4hcreset (DIS_CLR_CCS_4_HCRESET)**
29	0h RW	**disable_rawlfps_based_wake_fix (DISABLE_RAWLFPS_BASED_WAKE_FIX)**
28:7	0h RO	**Rsvd (Rsvd)**
6:0	0h RW	**SSPE_REG (SSPE_REG)**

16.228 (SSPITPE)—Offset 80BCh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 7Fh

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**RSVD (RSVD)**
6:0	7Fh RW	**ITP Transmit Enable (ITP_TRANSMIT_EN):** Width is scaled with USB3 Port Count PortCount=4

16.229 AUX Reset Control (AUX_CTRL_REG)—Offset 80C0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 15FC0F0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Reset USB3 AUX/Main PD Logic (RST_U3_AM_PDL):** A reset that is designed to reset all USB3 port AUX + Main power domain logic. This is a debug function. It is called cold reset. Write 1 to this register will issue a cold reset to USB3 ports.
30	0h RW	**RESERVED (RSVD)**
29	0h RW	**Enable PCIe PIPE Reset As PCIe PHY Reset (EN_PPIPE_PPHY_RST):** This bit set to 1 selects the PCIe PIPE reset as a PCIe PHY power on reset. 0 selects the AUX powerup reset as a PCIe PHY powerup reset.
28	0h RW	**Ensure PIPE Powerdown On PERST# (EN_PPWRDN_PERST):** This bit ensures any PERST# being asserted will cause PIPE powerdown state transition to P1.
27	0h RW	**Disable PERST# Main RST (DIS_PERST_MRST):** There is a feature where we only allow PERST# to be treated as a main powerdown reset when it is asserted during the state that is not in D3. This bit disables this feature when it is set to 1. If this bit is set to 1, it means that any PERST# will be treated as a main power domain reset regardless of its Dstate.
26	0h RW	**Disable PCIe PERST# Host Controller (DIS_PERST_HC):** There is a feature where we can disable PCIe PERST# from reset the host controller until power management control module has done its clock switching. This is for a case where PM entered L23 and immediately PERST# asserted. This bit is designed to enable this feature when set to 1.
25	0h RW	**Enable Fast Simulation Reset Mode (EN_FAST_RST):** Enable a fast simulation mode for reset function. This is a feature for GLS
24	1h RW	**Prevent USB3 Link Down Reset (PREV_U3_LDRST):** When set to 1 prevent a reset being generated due to the USB3 port link down condition.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**Ignore Main Power Up Reset (IGN_MPU_RST):** When set to 1 ignore the main power up reset for USB3 PIPE PHY reset.
22	1h RW	**RESERVED (RSVD_1)**
21	0h RW	**Ignore HC Reset USB2 (IGN_HC_RST_U2):** When set to '1' ignore HC reset to reset the USB2 Port logic
20	1h RW	**Ignore HC Reset USB PHY (IGN_HC_RST_UP_POR):** When set to '1' ignore HC reset to the USB PHY power-on reset
19	1h RW	**Enable PCIe Link-Down Reset (EN_PLD_RST):** Enable a reset due to a PCIe link-down condition. The PCIe link down condition will cause a HC reset liked. If this bit is set 1, the PCIe link down condition will only reset the PCIe core.
18	1h RW	**Enable EEPROM Reload On Power Up (EN_EEP_REL_PU):** When set to '1' enable EEPROM reload on every main power-up
17	1h RW	**Ignore HC Reset PCIe PHY PIPE (IGN_HC_RST_PPP):** When set to '1' ignore HC reset to the PCIe PHY PIPE reset
16	1h RW	**Ignore LTSSM Reset USB PHY PIPE (IGN_LRST_UPP):** When set to '1' ignore the LTSSM of USB link state transition caused reset to USB PHY PIPE reset
15	1h RW	**Ignore Warm Reset USB PHY Power (IGN_WR_UPP):** When set to '1' ignore warm reset the USB PHY power on reset
14	1h RW	**Allow Core PCIe Link Down Reset (ALL_CPLD_RST):** When set to '1' allow PCIe link down to cause a reset to the rest of the core
13	0h RW	**Ignore Hot Reset USB3 (IGN_HR_U3):** When set to '1' ignore hot reset to the USB3 port logic
12	0h RW	**Ignore Warm Reset USB3 (IGN_WR_U3):** When set to '1' ignore warm reset to the USB3 port logic
11	0h RW	**Ignore Main Power Up Reset USB3 (IGN_MPU_RST_U3):** When set to '1' ignore main power up reset to USB3 port logic
10	0h RW	**Ignore Main Power Up Reset USB2 (IGN_MPU_RST_U2):** When set to '1' ignore main power up reset to USB2 port logic
9	0h RW	**Ignore Main Power Up Reset PCIe Core (IGN_MPU_RST_PC):** When set to '1' ignore main power up reset to PCIe core
8	0h RW	**Ignore Main Power Up Reset PCIe PHY (IGN_MPU_RST_PP):** When set to '1' ignore main power up reset to PCIe PHY
7	1h RW	**Ignore HC Reset USB PHY (IGN_HC_RST_UP):** When set to '1' ignore HC reset to the USB PHY
6	1h RW	**Ignore Warm Reset USB PHY (IGN_WRST_UP):** When set to '1' ignore warm reset to the USB PHY
5	1h RW	**Enable HC Reset Per Port Isolation (EN_HC_RST_PPI):** Enables the HC reset or per port reset isolation function

Bit Range	Default & Access	Field Name (ID): Description
4	1h RW	**Allow Power Off Power Domain Reset (ALL_PO_PDRST):** When set to '1' allow main power off condition to trigger a main power domain reset
3	0h RW	**Ignore Wait For PERST# During Power Show Down (IGN_PERST_PSD):** When set to '1' ignore waiting for PERST# deassertion during main power show down.
2	0h RW	**Ignore Fundamental Reset During AUX Power Up (IGN_FRST_AUX_PU):** When fundamental reset is asserted during AUX power up, if this bit is set, then we will ignore PERST# such that purely wait for timeout to deassert fundamental reset.
1:0	0h RW	**Trigger Fundamental Reset (TRIG_FRST):** Writing to bit(1:0) to value of 2'b11 will cause a fundamental reset

16.230 Super Speed Bandwidth Overload (HOST_BW_OV_SS_REG)—Offset 80C4h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 4A4008h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**RESERVED (RSVD)**
23:12	4A4h RW	**Max. TT BW Allowed (MAX_TT_BWA):** see white paper
11:0	8h RW	**Per Packet Overhead SS BW (PP_OVRH_SSBW):** BW calculation: Overhead per packet for SS BW calculations. see white paper.

16.231 High Speed TT Bandwidth Overload (HOST_BW_OV_HS_REG)—Offset 80C8h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 1A01Fh

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**RESERVED (RSVD)**

Bit Range	Default & Access	Field Name (ID): Description
23:12	1Ah RW	**Per Packet Overhead HS-TT BW (PP_OVRH_HSTTBW):** BW calculation: Overhead per packet for HS-TT BW calculations. see white paper.
11:0	1Fh RW	**Per Packet Overhead HS BW (PP_OVRH_HSBW):** BW calculation: Overhead per packet for HS BW calculations. see white paper.

16.232 Bandwidth Overload Full Low Speed (HOST_BW_OV_FS_LS_REG)—Offset 80CCh

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 14080h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**RESERVED (RSVD)**
23:12	14h RW	**Per Packet Overhead FS BW (PP_OVRH_FSBW):** BW calculation: Overhead per packet for FS BW calculations. see white paper.
11:0	80h RW	**Per Packet Overhead LS BW (PP_OVRH_LSBW):** BW calculation: Overhead per packet for LS BW calculations. see white paper.

16.233 System Bandwidth Overload (HOST_BW_OV_SYS_REG)—Offset 80D0h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 32010h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**RESERVED (RSVD)**
23:12	32h RW	**Per TT Packet Overhead System BW (PTTP_OVRH_SBW):** BW calculation: Overhead per TT packet for System BW calculations. see white paper.

Bit Range	Default & Access	Field Name (ID): Description
11:0	10h RW	**Per Packet Overhead System BW (PP_OVRH_SBW):** BW calculation: Overhead per packet for System BW calculations. see white paper.

16.234 Scheduler Async Delay (HOST_CTRL_SCH_ASYNC_DELAY_REG)—Offset 80D4h

Global defaults for inserting delays between packets in the scheduler for async. types.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RO	**RESERVED (RSVD)**
19	0h RW	**High-Speed Bulk Delay Enable (HS_BD_EN)**
18:16	0h RW	**High-Speed Bulk Delay Default (HS_BD_DEF):** (0=125us,1=250us,2=500us,3=1ms,...)
15	0h RW	**Full-Speed Bulk Delay Enable (FS_BD_EN)**
14:12	0h RW	**Full-Speed Bulk Delay Default (FS_BD_DEF):** (0=125us,1=250us,2=500us,3=1ms,...)
11	0h RW	**High-Speed Control Delay Enable (HS_CD_EN)**
10:8	0h RW	**High-Speed Control Delay Default (HS_CD_DEF):** (0=125us,1=250us,2=500us,3=1ms,...)
7	0h RW	**Full-Speed Control Delay Enable (FS_CD_EN)**
6:4	0h RW	**Full-Speed Control Default (FS_CD_DEF):** (0=125us,1=250us,2=500us,3=1ms,...)
3	0h RW	**Low-Speed Control Delay Enable (LS_CD_EN)**
2:0	0h RW	**Low-Speed Control Delay Default (LS_CD_DEF):** (0=125us,1=250us,2=500us,3=1ms,...)

16.235 DEVICE MODE CONTROL REG 0 (DUAL_ROLE_CFG_REG0)—Offset 80D8h

All bits in this register must be in the Always ON Power domain (ungated SUS or AON as appropriate)

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 800h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**XDCI_DRD_CTRL_EN (REG_IN_XDCI):** When DRD_ACCESS_MODE (bit 18) =1, 0 -- The policies below is controlled by register in the XHCI MMIO 1 -- The policies below is controlled by register in the XDCI MMIO When DRD_ACCESS_MODE =0, this bit is a don?t care.
30:28	0h RW	**DRD_WDT_TIMER (DRD_WDT_TIMER):** 0-based incremental of 500ms
27	0h RW	**EN_DRD_WDT_SWITCH_FROM_HOST_TO_DEVICE (EN_DRD_WDT_SWITCH_FROM_HOST_TO_DEVICE)**
26	0h RW	**EN_DRD_WDT_SWITCH_FROM_DEVICE_TO_HOST (EN_DRD_WDT_SWITCH_FROM_DEVICE_TO_HOST)**
25	0h RW	**EN_DRD_WDT_SWITCH_OUT_OF_COLD_RESET (EN_DRD_WDT_SWITCH_OUT_OF_COLD_RESET)**
24	0h RW	**SW_VBUS_VALID (SW_VBUS_VALID):** if SW_IDPIN_EN (bit 21) is 1, 0 ? deassert sw vbus valid 1 ? assert sw vbus valid
23	0h RW	**EN_PIPE_4_1_SYNC_PHY_STATUS (EN_PIPE_4_1_SYNC_PHY_STATUS)**
22	0h RW	**EN_PIPE_RX_ON_IDPIN (EN_PIPE_RX_ON_IDPIN):** During the connection to a device, there may be a delay in DRD switch from XDCI to XHCI mode,the rx term can be low after idpin deasserts. If this bit is 0, the rx term will be assert immediately after idpin toggle. Otherwise, the device may fall back to USB2 mode. 1 -- drive 0s on utmi rx signals to controller if not connected.
21	0h RW	**SW_IDPIN_EN (SW_IDPIN_EN):** SW_IDPIN_EN 1 -- enable SW id pin (pre DRD) 0 -- disable SW id pin.
20	0h RW	**SW_IDPIN (SW_IDPIN):** if SW_IDPIN_EN (bit 21) is 1, 0 - host mode 1 - device mode
19	0h RW	**USB2_SUSP_OR_DIS (USB2_SUSP_OR_DIS):** 1 - The DRD UTMI suspendm will be controlled by host/device based on the DRD switch. 0 - whenever device or host deassert suspendm, the DRD UTMI suspendm will be deasserted.
18	0h RW	**DRD_ACCESS_MODE (DRD_ACCESS_MODE):** 0 - The DUAL_ROLE register from host and device are ORed to control DRD 1 - The DUAL_ROLE register from host and device are selected by XDCI_DRD_CTRL_EN (bit 31) to control DRD
17	0h RW	**EN_DIRECT_DRD_SWITCH_ON_USB3PORT_BY_IDPIN (EN_DIRECT_DRD_SWITCH_ON_USB3PORT_BY_IDPIN):** 0 enable the direct DRD switch on USB3 port by idpin 1 disable the direct DRD switch

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW	**SW_SWITCH_ENABLE (SW_SWITCH_ENABLE):** SW switch enable 0 (default) ID pin HW controlled DRD. 1 -- SW controlled DRD (ignore idpin), switch based on the DRD_CONFIG.
15	0h RW	**RSVD (RSVD)**
14:3	100h RW	**DEBOUNCE_VAL (DEBOUNCE_VAL):** ID ping debounce timer (DEBOUNCE_VAL): in the unit of RTC clock (33us) default to 8.448 ms
2	0h RW	**SYNCHRONIZE_SS_HS_SWITCH (SYNCHRONIZE_SS_HS_SWITCH):** Synchronize the SS and HS switch: 0 (default) Does not synchronize. i.e. HS switch on the debounced id pin, while SS switch independently controlled by the sequencer. 1 synchronize HS and SS switch. Both speeds switch when sequencer switch.
1:0	0h RW	**DRD_CONFIG (DRD_CONFIG):** 00 Dynamic DRD switch mode 01 static host mode 10 static device mode 11 -- reserved

16.236 DEVICE MODE CONTROL REG 1 (DUAL_ROLE_CFG_REG1)—Offset 80DCh

All bits in this register must be in the Always ON Power domain (ungated SUS or AON as appropriate)

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 10000000h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**BVALID_HW (HW_BVALID):** hardware bvalid
30	0h RO	**DRD_STATUS_B31_30 (DRD_STATUS_30)**
29	0h RO	**DRD_MODE (DRD_MODE):** SS DRD mode 0 device mode 1 host mode
28	1h RO	**IDPIN (IDPIN):** IDPIN value hw version
27:16	0h RO	**DRD_STATUS_B27_16 (DRD_STATUS_B27_16)**
15:0	0h RW	**DRD_CONTROL_B15_0 (DRD_CONTROL_B15_0)**

16.237 AUX Power Management Control (AUX_CTRL_REG1)—Offset 80E0h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 808DBCA0h

Bit Range	Default & Access	Field Name (ID): Description
31	1h RW	**D3 Hot function enable register (D3_HOT_FXN_EN):** This bit is from pin input which is set 1. But we allow software to alter it if it is needed. 1: D3 hot enabled 0: D3 hot not abled.
30	0h RW	**Allow L1 Core Clock Gating (ALL_L1_CORE_CG):** When set to 1 allows core clock being gated during L1 state.
29	0h RW	**Allow Engine PHY Status Extension (AL_EP_SEXT):** When set to 1 allows the engine to extend PHY status of PCIe PIPE for one more cycle. This is due to the fact that our rate change function has a potential of not being able to sample the phystatus signal.
28	0h RW	**Allow Engine PCIe Rate Change Passing (ALL_EP_RCP):** When set to 1 allows the engine to pass PCIe rate change signal as it is from PCIe core to PCIe PHY.
27	0h RW	**Allow Engine PERST Fundamental Reset (AL_PERST_FRST):** When set to 1 allow engine to treat PERST# as a foundamental reset
26	0h RW	**Overwrite PCIe P2 to P1 (OVR_PCIE_P2_P1):** When set to 1 will overwrite a PCIe powerdown state of P2 to P1.
25	0h RW	**Set Internal SSV 1 (SET_ISSV_1):** When set to 1 set the internal SSV to 1.
24	0h RW	**Clear Internal SSV 0 (CLR_ISSV_0):** When set to 1 clear the internal SSV to 0.
23	1h RW	**Enable save_restore_enable SW Loading (EN_SRE_SW_LD):** This is a bit that enables the save_restore_enable signal being loaded when a software command has set Save bit. This is a debug function.
22	0h RW	**RESERVED (RSVD_1)**
21	0h RW	**Force save_restore 1 (FORCE_SR1):** When set to 1, it will force the save_restore flag to 1. This flag is an bit to ensure that we have masked the update during low power state. If software write this bit to 1, it must write it to 0 in order to resume the normal save and restore function.
20	0h RW	**CFG DISABLE_WARM_RST_DET_specUpPorts (cfg_dis_WrstDet_specU)**
19	1h RW	**cfg iob drivestrength[1] (CIDS1)**

Bit Range	Default & Access	Field Name (ID): Description
18	1h RW	**cfg iob drivestrength[0] (CIDS0)**
17	0h RW	**Enable CFG USB P2 (EN_CFG_UP2):** When set to '1' enable cfg usb p2
16	1h RW	**cfg clk gate dis (CCGD)**
15	1h RW	**Enable CFG RXDET P3 (EN_CFG_RDP3):** When set to '1' enable cfg rxdet p3
14	0h RW	**Enable CFG PIPE Reset (EN_CFG_PIPE_RST):** When set to '1' enable cfg pipe rst
13	1h RW	**Enable Filter TX Idle (EN_FILT_TX_IDLE):** When set to 1 enables a filter function to TX electrical idle signal at PCIe PIPE. We have a filter that will set TXelecidle signal of PCIe PIPE to 1 whenever we are in isolation state or power down transition states.
12	1h RW	**Enable Host Engine Generate PME (EN_HE_GEN_PME):** This is a global switch to whether or not eable this host engine to generate PME message.
11	1h RW	**Enable Isolation (EN_ISOL):** When set to '1' enable isolation
10	1h RW	**Enable L1 Caused P2 Overwrite (EN_L1_P2_OVR):** Set 1 to enable a new feature. This new feature is designed to use L1 as a state to identify whether we should do P2 Overwrite or not. We used to use P1 state to identify whether or not to invoke P2 overwrite function.
9	0h RW	**Enable Core Clock Gating (EN_CORE_CG):** When set to '1' enable core clock gating based on low power state entered
8	0h RW	**Enable PHY Status Timeout (EN_PHY_STS_TO):** When set to '1' enable PHY status timeout function which is designed to cover the PCIePHY issue that we may have not able to detect the PHY status toggle.
7	1h RW	**Ignore aux_pm_en PCIe Core (IGN_APE_PC):** When set to '1' ignore the aux_pm_en reg from PCIe core to continue the remote wake/clock switching support
6	0h RW	**Enable P2 Overwrite P1 (EN_P2_OVR_P1):** When set to '1' enable P2 overwrite P1 when PCIe core has indicated the transition from P0 to P1. This is to enable entering the even lower power state.
5	1h RW	**Enable P2 Remote Wake (EN_P2_REM_WAKE):** When set 1 '1' enable the remote wake function by allowing P2 clock/switching and P2 entering
4:1	0h RW	**Forced PM State (FORCED_PM_STATE)**
0	0h RW	**Initiate Force PM State (INIT_FPMS):** When set to '1' force PM state to go to the state indicated in bit 4:1

16.238 Battery Charge (BATTERY_CHARGE_REG)—Offset 80E4h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Enable DM_SRC Battery Charge (EN_DS_BC):** 1 - Always enable battery charge DM_SRC if not connected. Don't wait for portable device detect. (spec. ver. 1.2) 0 - Battery charge spec ver. 1.1.
30:4	0h RO	**RESERVED (RSVD)**
3	0h RW	**Enable Port 3 Battery Charging (EN_P3_BC):** 0 - Battery charging disabled (Physical Port #3) 1 - Battery charging enabled (Physical Port #3)
2	0h RW	**Enable Port 2 Battery Charging (EN_P2_BC):** 0 - Battery charging disabled (Physical Port #2) 1 - Battery charging enabled (Physical Port #2)
1	0h RW	**Enable Port 1 Battery Charging (EN_P1_BC):** 0 - Battery charging disabled (Physical Port #1) 1 - Battery charging enabled (Physical Port #1)
0	0h RW	**Enable Port 0 Battery Charging (EN_P0_BC):** 0 - Battery charging disabled (Physical Port #0) 1 - Battery charging enabled (Physical Port #0)

16.239 Port Watermark (HOST_CTRL_WATERMARK_REG)—Offset 80E8h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 800080h

Bit Range	Default & Access	Field Name (ID): Description
31:16	80h RW	**RBUF water mark (RBUF_WM)**
15:0	80h RW	**XBUF water mark (XBUF_WM)**

16.240 SuperSpeed Port Link Control (HOST_CTRL_PORT_LINK_REG)—Offset 80ECh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 18010000h

Bit Range	Default & Access	Field Name (ID): Description
31:27	3h RW	**Force LTSSM State (FORCE_LTSSM_ST):** LTSSM state to be force.
26	0h RW	**Direct Link LTSSM State (DL_LTSSM_ST):** 0: Normal operation mode 1: Direct link to a specific state specified by bit 31:27. It shall be written 0 in normal operation mode.
25	0h RW	**Direct Link To U0 (DL_U0):** 0: Normal operation mode 1: Direct link to U0. It shall be written 0 in normal operation mode.
24:21	0h RW	**Forced Compliance Pattern (FORCED_CMP_PAT):** Compliance pattern to be forced to enter compliance mode.
20	0h RW	**Enable Link Error Slave Count (EN_LES_CNT):** 0: Disable link error slave count 1: Enable link error slave count
19:17	0h RW	**Debug Mode Select (DEBUG_MD_SEL)**
16:15	2h RW	**PHY Low Power Latency (PHY_LP_LAT):** This field defines the latency to drive the PHY to enter low power mode 0: 4 cycles 1: 8 cycles 2: 16 cycles 3: 32 cycles
14:12	0h RW	**Link Recovery Minimum Time (LR_MIN_TM):** This value defines the minimum time for the link to stay in Recovery.Active other than from U3. The granuity is 128us.
11:9	0h RW	**Link Polling Minimum Time (LP_MIN_TM):** This value defines the minimum time for the link to stay in Polling.Active and Recovery.Active from U3. The granuity is 128us.
8	0h RW	**Force Link Accept PM Command (FORCE_LA_PMC):** 0: Normal operation mode 1: Force link to accept power management command
7	0h RW	**Direct Link Recovery U0 (DL_REC_U0):** 0: Normal operation mode 1: Direct link to Recovery from U0
6	0h RW	**Link Fast Training Mode (LINK_FTM):** 0: Normal operation mode 1: Link fast training mode This bit should be written 0 in normal operation.
5	0h RW	**Disable Link Scrambler (DIS_LINK_SCRAM):** 0: Enable link scrambler 1: Disable link scramber
4	0h RW	**Direct Link U3 From U0 (DL_U3_U0):** 0: Normal operation mode 1: Direct link to U3 from U0. It shall be written 0 in normal operation mode.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**Direct Link U3 From U0 (DL_U2_U0):** 0: Normal operation mode 1: Direct link to U2 from U0 . It shall be written 0 in normal operation mode.
2	0h RW	**Direct Link U3 From U0 (DL_U1_U0):** 0: Normal operation mode 1: Direct link to U1 from U0 . It shall be written 0 in normal operation mode.
1	0h RW	**Enable Link Loopback Master Mode (EN_LINK_LB_MAST):** 0: Disable link loopback master mode 1: Enable link loopback master mode
0	0h RW	**Disable Link Compliance Mode (DIS_LINK_CM):** 0: Enable link compliance mode 1: Disable link compliance mode

16.241 USB2 Port Link Control 1 (USB2_LINK_MGR_CTRL_REG1)—Offset 80F0h

These set of registers is used to control jey USB set of timers. They are spread over 4 registers each 32 bits wide.

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 310803A0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	31h RW	**FS/LS Mode SE0 Disconnect Delay[7:0] (FSLS_SE0_DIS_DEL_7_0):** # of microseconds of SE0 in FS/LS mode to register disconnect had occurred.
23:21	0h RW	**Reserved (RSVD0)**
20	0h RW	**L1_EXIT_RECOVERY_MODE (L1_EXIT_RECOVERY_MODE):** Mode for extended L1 Exit recovery delay: 0: 12us 1: 50us
19	1h RW	**L1_TO_INCR_MODE (L1_TO_INCR_MODE):** Mode select for L1 Timeout increments: 0: time out increments are in 125us 1: L1 Timeout increments are in 256us. Refer to USB2 PORTHLPMC.L1 Timeout in XHCI Spec for additional details
18	0h RW	**Reserved (RSVD1)**
17	0h RW	**EN_DETECT_NOMINAL_PKT_EOP (EN_DETECT_NOMINAL_PKT_EOP):** 0: Detect minimal packet EOP. 1: Detect nominal packet EOP.
16	0h RW	**Disable Chirp Response (DIS_CHIRP_RESPONSE):** 0: Normal 1: Force full speed on host ports (disable chirp response)

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**Disable 192 Byte Limit Check (DIS_192B_LIM):** 0: Enforce 192 byte limit on complete-split INs. Treat any packet) 192 as babble case. 1: Disable 192 byte limit check.
14	0h RW	**External Provided FS/LS Disconnect (EXT_FSLS_DIS):** 0: Internal FS/LS Disconnect from linestate(1:0) 1: External provided FS/LS Disconnect from hostdisconnect input
13:12	0h RW	**UTMI Reset Source Select (UTMI_RST_SEL):** Select UTMI Reset Source (FRD UTMI Reset Only) 00: HCReset or Force PHY Reset or internal reset after disconnect/suspend for restart (default) 01,11: UTMI reset = ~UTMI suspendm 10: UTMI reset = ~UTMI suspendm and synchronization to port clk.
11	0h RW	**Disable HS Disconnect Window (DIS_HS_DIS_WIN):** 0: Enable HS Disconnect Window Function 1: Disable HS Disconnect Window Function
10	0h RW	**Disable Port Error Detection (DIS_PERR_DET):** 0: Enable Port Error Detection (default) 1: Disable Port Error Detection
9	1h RW	**Disable Peek Function for ISO-OUT (DIS_PF_IOUT):** 0: Enable Peek function for ISO-OUT (default) 1: Disable Peek function for ISO-OUT
8	1h RW	**Drive Resume-K FS/LS Serial Interface (DRV_RESK_FSLS_SER):** 0: Drive Resume-K on parallel Interface 1: Drive Resume-K directly on FS/LS Serial Interface (default)
7	1h RW	**Enable USB2 Drop-Ping (EN_U2_DROP_PING):** 0: Disable Drop-Ping Function in USB2 Protocol (default) 1: Enable Drop-Ping Function in USB2 Protocol
6	0h RW	**Enable USB2 Force-Ping (EN_U2_FORCE_PING):** 0: Disable Force-Ping Function in USB2 Protocol (default) 1: Enable Force-Ping Function in USB2 Protocol
5	1h RW	**Enable USB2 Auto-Ping (EN_U2_AUTO_PING):** 0: Disable Auto-Ping Function 1: Enable Auto-Ping Function in USB2 Protocol (default)
4	0h RW	**Disable PHY SuspendM (DIS_PHY_SUSM):** 0: PHY is suspend=U3,U2,disconnect (default) 1: Disable PHY SuspendM in All States
3	0h RW	**UTMI Internal Clock Gate Disable (UTMI_INT_CG_DIS):** 0: Normal operation (internal clock gated in U2,U3,disconnect) 1: UTMI Internal Clock Gate Disable
2	0h RW	**Disable PHY SuspendM in Disconnect State (DIS_PSUSM_DS):** 0: PHY is suspendM=0 in Disconnect State (default) 1: Disable PHY SuspendM in Disconnect State
1	0h RW	**Force PHY Reset (FORCE_PHY_RST):** 0: Normal Operation (default) 1: Force PHY Reset
0	0h RW	**USB2 Accelerated Simulation Timing (U2_ACC_SIM_TIM):** 0: Normal Operation (default - FPGA/ASIC) 1: USB2 Accelerated Simulation Timing (default - simulation)

16.242 USB2 Port Link Control 2 (USB2_LINK_MGR_CTRL_REG2)—Offset 80F4h

These set of registers is used to control jey USB set of timers. They are spread over 4 registers each 32 bits wide.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 80C40620h

Bit Range	Default & Access	Field Name (ID): Description
31	1h RW	**Total Reset Duration[0] (TOT_RST_DUR_0):** # of microseconds for total reset duration
30:18	31h RW	**Chirp-K Duration (CHIRPK_DUR):** # of microseconds of Chirp-K to register that a device is chirping
17:5	31h RW	**K/J Disconnect Connect Delay (KJ_DIS_CON_DEL):** # of microseconds of K/J in disconnected state to register connect has occurred.
4:0	0h RW	**FS/LS Mode SE0 Disconnect Delay[12:8] (FSLS_SE0_DIS_DEL_12_8):** # of microseconds of SE0 in FS/LS mode to register disconnect had occurred.

16.243 USB2 Port Link Control 3 (USB2_LINK_MGR_CTRL_REG3)—Offset 80F8h

These set of registers is used to control jey USB set of timers. They are spread over 4 registers each 32 bits wide.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: F865EB6Bh

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RW	**U2 Entry Ignore Linestate Changes Duration[3:0] (U2_IGN_LS_DUR_3_0):** # of microseconds after entering U2, linestate changes are ignored as bus settles
27:15	10CBh RW	**U3 Entry Ignore Linestate Changes Duration (U3_IGN_LS_DUR):** # of microseconds after entering U3, linestate changes are ignored as bus settles
14:0	6B6Bh RW	**Total Reset Duration[15:1] (TOT_RST_DUR_15_1):** # of microseconds for total reset duration

16.244 USB2 Port Link Control 4 (USB2_LINK_MGR_CTRL_REG4)—Offset 80FCh

These set of registers is used to control jey USB set of timers. They are spread over 4 registers each 32 bits wide.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 8003h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**RESERVED (RSVD)**
21:9	40h RW	**U2 Detect Remote Wake Delay (U2D_RWAKE_DEL):** #of microseconds after detecting U2 remote wake condition to reflect K
8:0	3h RW	**U2 Entry Ignore Linestate Changes Duration[12:4] (U2_IGN_LS_DUR_12_4):** # of microseconds after entering U2, linestate changes are ignored as bus settles

16.245 Bandwidth Calc Control (HOST_CTRL_BW_CTRL_REG)—Offset 8100h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 8008h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD)**
15:0	8008h RW	**Reserved (RSVD_1)**

16.246 Host Interface Control (HOST_IF_CTRL_REG)—Offset 8108h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RW	**Rsvd1 (Rsvd1)**
0	1h RW	**Host IF (HOSTIF)**

16.247 Bandwidth Overload Burst (HOST_BW_OV_BURST_REG)—Offset 810Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 8020h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**RESERVED (RSVD)**
23:12	8h RW	**Per Burst Overhead System BW (PB_OVRH_SBW):** BW calculation: Overhead per burst for system BW calculations. see white paper.
11:0	20h RW	**Per Burst Overhead System BW (PB_OVRH_SSBW):** BW calculation: Overhead per burst for SS BW calculations. see white paper.

16.248 USB Max Bandwidth Control 4 (HOST_CTRL_BW_MAX_REG)—Offset 8128h

Access Method

Type: MEM Register
(Size: 64 bits)

Device:
Function:

Default: F42528505647F42h

Bit Range	Default & Access	Field Name (ID): Description
63:60	0h RO	**Reserved (RSVD)**
59:48	F42h RW	**PCIe Max BW Units (PCIE_MAX_BW):** Max. Number of BW units for PCIe (system interface) (denominator in 90% calculation)
47:36	528h RW	**TT Max BW Units (TT_MAX_BW):** Max. Number of BW units for TTs. (denominator in 90% calculation)
35:24	505h RW	**FS/LS Max BW Units (FSLS_MAX_BW):** Max. Number of BW units for FS/LS ports. (denominator in 90% calculation)

Bit Range	Default & Access	Field Name (ID): Description
23:12	647h RW	**HS Max BW Units (HS_MAX_BW):** Max. Number of BW units for HS ports. (denominator in 80% calculation)
11:0	F42h RW	**SS Max BW Units (SS_MAX_BW):** Max. Number of BW units for SS ports. (denominator in 90% calculation)

16.249 USB2 Linestate Debug (LINESTATE_DEBUG_REG)—Offset 8130h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**RESERVED (RSVD)**
27:26	0h RO	**Port 14 UTMI Linestate (P14_UTMI_LS)**
25:24	0h RO	**Port 13 UTMI Linestate (P13_UTMI_LS)**
23:22	0h RO	**Port 12 UTMI Linestate (P12_UTMI_LS)**
21:20	0h RO	**Port 11 UTMI Linestate (P11_UTMI_LS)**
19:18	0h RO	**Port 10 UTMI Linestate (P10_UTMI_LS)**
17:16	0h RO	**Port 9 UTMI Linestate (P9_UTMI_LS)**
15:14	0h RO	**Port 8 UTMI Linestate (P8_UTMI_LS)**
13:12	0h RO	**Port 7 UTMI Linestate (P7_UTMI_LS)**
11:10	0h RO	**Port 6 UTMI Linestate (P6_UTMI_LS)**
9:8	0h RO	**Port 5 UTMI Linestate (P5_UTMI_LS)**
7:6	0h RO	**Port 4 UTMI Linestate (P4_UTMI_LS)**
5:4	0h RO	**Port 3 UTMI Linestate (P3_UTMI_LS)**
3:2	0h RO	**Port 2 UTMI Linestate (P2_UTMI_LS)**
1:0	0h RO	**Port 1 UTMI Linestate (P1_UTMI_LS)**

16.250 USB2 Protocol Gap Timer (USB2_PROTOCOL_GAP_TIMER_REG)—Offset 8134h

Access Method

Type: MEM Register (Size: 64 bits)

Device:
Function:

Default: C3C640C05140Ch

Bit Range	Default & Access	Field Name (ID): Description
63:56	0h RO	**Reserved (RSVD)**
55:48	Ch RW	**GAP time after LS TX thru FS hub (LSTX_GAP_TIME)**
47:40	3Ch RW	**GAP time after LS RX thru FS hub (LSRX_GAP_TIME)**
39:32	64h RW	**GAP timer after LS (LS_GAP_TIME)**
31:24	Ch RW	**GAP time after FS (FS_GAP_TIME)**
23:16	5h RW	**GAP time after HS RX (HSRX_GAP_TIME)**
15:8	14h RW	**GAP time after HS TX SOF (HSTXSOF_GAP_TIME)**
7:0	Ch RW	**GAP time HS TX Packet (HSTX_GAP_TIME)**

16.251 USB2 Protocol Bus Timeout Timer (USB2_PROTOCOL_BTO_TIMER_REG)—Offset 813Ch

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 8D4258B8h

Bit Range	Default & Access	Field Name (ID): Description
31:21	46Ah RW	**Bus timeout count for LS (LS_BUS_TO)**
20:10	96h RW	**Bus timeout count for FS (FS_BUS_TO)**
9:0	B8h RW	**Bus timeout count for HS (HS_BUS_TO)**

16.252 Power Scheduler Control-0 (PWR_SCHED_CTRL0)—Offset 8140h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: A019132h

Bit Range	Default & Access	Field Name (ID): Description
31:24	Ah RW	**Engine Idle Hysteresis (EIH):** This register controls the min. idle span that has to be observed from the engine idle indicators before the power state flags (xhc_*_idle) will indicate a 1.
23:12	19h RW	**Backbone PLL Shutdown Advance Wake (BPSAW):** This register controls the time before the next scheduled transaction where the Backbone PLL request will assert. Register Format: Bits [11:7] # of 125us uframes Bits [6:0] # of microseconds (0-124)
11:0	132h RW	**Backbone PLL Shutdown Min. Idle Duration (BPSMID):** The sum of this register plus the Backbone PLL Shutdown Advance Wake form to a Total Idle time. When the next scheduled periodic transaction is after present time + Total Idle, the Backbone PLL request will de-assert, allowing the PLL to shutdwon. Register Format: Bits [11:7] # of 125us uframes Bits [6:0] # of microseconds (0-124)

16.253 Power Scheduler Control-2 (PWR_SCHED_CTRL2)—Offset 8144h

These bit enable by EP type those EPs classes that are considered for determining next periodic active interval for pre-wake of the periodic_active signal. EP classes that are disabled may never be observed in setting of the periodic_active signal.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 33Fh

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RO	**Reserved (RSVD)**
9	1h RW	**HS Interrupt-OUT Alarm (HS_INT_OUT_ALRM)**
8	1h RW	**HS Interrupt-IN Alarm (HS_INT_IN_ALRM)**
7	0h RW	**SS Interrupt-OUT FC Alarm (SS_INT_OUT_FC_ALRM)**

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**SS Interrupt-IN Alarm (SS_INT_IN_FC_ALRM)**
5	1h RW	**SS Interrupt-OUT & not in FC Alarm (SS_INT_OUT_ALRM)**
4	1h RW	**SS Interrupt-IN & not in FC Alarm (SS_INT_IN_ALRM)**
3	1h RW	**HS ISO-OUT Alarm (HS_ISO_OUT_ALRM)**
2	1h RW	**HS ISO-IN Alarm (HS_ISO_IN_ALRM)**
1	1h RW	**SS ISO-OUT Alarm (SS_ISO_OUT_ALRM)**
0	1h RW	**SS ISO-IN Alarm (SS_ISO_IN_ALRM)**

16.254 AUX Power Management Control (AUX_CTRL_REG2)—Offset 8154h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 81390206h

Bit Range	Default & Access	Field Name (ID): Description
31	1h RW	**DIS_L1P2_EXIT_ON_WAKE_EN (DIS_L1P2_EXIT_ON_WAKE_EN):** This bit disables the dependency on Wake Enables defined in PORTSC for L1P2 exit when in D0
30:28	0h RW	**RESERVED0 (RSVD0)**
27:25	0h RW	**RESERVED (RSVD)**
24	1h RW	**Enable L1 exit notification to SNPS PCIe core (EN_L1_EXIT_NOTIF_PCIE):** This bit enables a L1 exit notification to SNPS PCIe core. There is a case where USB ports have waked up and AUX PM module has started the wakeup process. The AUX PM control state got into a wait for P0 state because it needs to wait until PCIe core to signal powerdown state change. Due to the fact that the core is in D3Hot, there is no run_stop bit set such that no internal interrupt will be fired. This causes the LTSSM of PCIe stayed in L1 even though AUX PM has known that it needs an L1 exit. This bit works together with bit21 of this register. 1: enables this feature 0: disables this feature.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**DISABLE PLC ON DISCONNECT (DIS_PLC_ON_DISCONNECT):** 1: do not assert PLC for disconnection 0: assert PLC for disconnection
22	0h RW	**TREAT_IDLE_AS_TS2_IN_LTSSM_WAIT_4_TS2 (TREAT_IDLE_AS_TS2_IN_LTSSM_WAIT_4_TS2):** This bit enables a feature in PCIe core LTSSM to treat IDLE received as TS2 when LTSSM is in wait for TS2 receive state. This is a function requested from PHY where it is possible to not able to receive TS2 without error. 1: treat Logic IDLE as TS2 received when in some PCIe LTSSM state. 0: disable this feature.
21	1h RW	**Disable p2 overwrite due to the D3HOT where PCIe core enters the L1 (DIS_P2_OVERWRITE_DUE2_D3HOT):** We added a feature where if PCIe core LTSSM enters L1 due to the D3hot, the aux PM control will not start a P2 overwrite function in anticipating for the next L23 enter. 1: disables p2 overwrite due to the D3HOT where PCIe core enters the L1. 0: enables P2 overwrite even if we are in D3Hot.
20	1h RW	**Enable the port to enter U3 automatically when in U1/U2 (ENABLE_AUTO_U3_ENTRY_FROM_U2_U3):** 1: enables the port to enter U3 automatically when in U1/U2 0: disables the port to enter U3 automatically when in U1/U2
19	1h RW	**No linkdown reset is issue during low power state (DIS_LINKDOWN_RST_DURING_LOW_POWER):** No linkdown reset is issue during low power state
18	0h RW	**EN_EXIT_DEEP_SLEEP_IF_PCIE_IN_P0 (EN_EXIT_DEEP_SLEEP_IF_PCIE_IN_P0):** This bit enables a feature in AUX PM module where if PCIe core LTSSM is in P0 for a duration of time, we will exit the deep sleep state. This is for failure control in case. 1: enables this feature 0: disable this feature
17	0h RW	**U2_EXIT_LFPS_TIMER_VALUE (U2_EXIT_LFPS_TIMER_VALUE):** This bit selects U2 exit LFPS timer value 0: 320ns 400ns in 25MHz domain 1: 240ns 320ns in 25MHz domain
16	1h RW	**EN_EXIT_DEEP_SLEEP_ON_USB_PORT_WAKEUP (EN_EXIT_DEEP_SLEEP_ON_USB_PORT_WAKEUP):** This bit enables a function that AUX PM module exits from the deep sleep state due to the USB ports wakeup level signal. We have added this feature where USB ports will generated a wakeup level signal to wakeup the AUX PM module if it is in deep sleep state and this level signal will be cleared if the change bits are updated by software. 1: enables this function 0: disables this function which means that a wakeup pulse generated from each USB PortSC event will wake up the AUX PM module from deep sleep if the D3 state is programmed.
15:14	0h RW	**P3_ENTRY_TIMEOUT (P3_ENTRY_TIMEOUT):** This field defines the timeout value to enter P3 mode in U2. 00: 7us 8us 01: 511us 512us 10: disables the timer (0us) 11: disables the timer (0us)

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**Enable U2 P3 Mode (EN_U2_P3):** 0: Disable U2 P3 mode 1: Enable U2 P3 mode
12:11	0h RW	**Fine Debug Mode Select (FINE_DM_SEL)**
10	0h RW	**Enable Low Power State Based Core Clock Gating (EN_LP_CORE_CG):** When set to '1' enable core clock gating based on low power state entered
9	1h RW	**Disable USB3 Port Status Changed Event (DIS_U3_PORT_SCE):** 0: Enable USB3 port status change event generation if any change bit is not cleared 1: Disable USB3 port status change event generation if any change bit is not cleared Bit 12 default 0
8:4	0h RW	**Debug Mode Select Register (DEB_MODE_SEL)**
3	0h RW	**Enable Auto Wakeup Non-IDLE (EN_AWAK_NIDLE):** When set to 1 enables the auto wakeup function when engine has identified non IDLE condition.
2	1h RW	**Enable PM Control P1 Exit P2 (EN_PMC_P1_EXIT_P2):** When set 1 enables the PM control module to transition to P1 instead of P0 when exit P2.
1	1h RW	**Enable PCIe PIPE CLK Isolation (EN_PP_CLK_ISOL):** When set to 1 enables the PCIe PIPE CLK to be isolated when main power is removed.
0	0h RW	**Enable P2 Overwrite P1 Allowed Detect (EN_P2OVRP1_ADET):** When set to 1 enables a function that can detect whether or not enable P2 overwrite P1 function. The condition to get to P2 overwrite is when engine is in idle conditions. This means that there is no ISO EP pending.

16.255 USB2 PHY Power Management Control (USB2_PHY_PMC)—Offset 8164h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: FCh

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD)**
7	1h RW	**EN_CMDM_TXRXB (EN_CMDM_TXRXB):** Enable Command Manager Active indication for Tx/Rx Bias circuit HS Phy PM Policy
6	1h RW	**EN_TTE_TXRXB (EN_TTE_TXRXB):** Enable TTE Active indication for Tx/Rx Bias circuit HS Phy PM Policy

Bit Range	Default & Access	Field Name (ID): Description
5	1h RW	**EN_IDMA_TXRXB (EN_IDMA_TXRXB):** Enable IDMA Active indication for Tx/Rx Bias circuit HS Phy PM Policy
4	1h RW	**EN_ODMA_TXRXB (EN_ODMA_TXRXB):** Enable ODMA Active indication for Tx/Rx Bias circuit HS Phy PM Policy
3	1h RW	**EN_TRM_TXRXB (EN_TRM_TXRXB):** Enable Transfer Manager Active indication for Tx/Rx Bias circuit HS Phy PM Policy
2	1h RW	**EN_SCH_TXRXB (EN_SCH_TXRXB):** Enable Scheduler Active indication for Tx/Rx Bias circuit HS Phy PM Policy
1	0h RW	**Enable Rx Bias ckt disable (EN_RXB_CD):** When set enables the Rx bias ckt to be disabled when conditions met (as described by the HS phy PM policy bits)
0	0h RW	**Enable Tx Bias ckt disable (EN_TXB_CD):** When set enables the Tx bias ckt to be disabled when conditions met (as described by the HS phy PM policy bits)

16.256 USB Power Gating Control (USB_PGC)—Offset 8168h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 282EEh

Bit Range	Default & Access	Field Name (ID): Description
31:1	14177h RO	**Reserved (RSVD)**
0	0h RW	**USB SRAM power gating enable (USB_SRAM_PGE):** When set enables power gating on USB ports. Usage of this bit is further qualified with xHCI SRAM Dynamic Power Gating Disable fuse. If the fuse disables dynamic power gating, setting this bit to 1 shall not enable power gating feature. This bit always returns the value that was written to it irrespective of the setting of xHCI SRAM Dynamic Power Gating Disable fuse.

16.257 xHCI Aux Clock Control Register (XHCI_AUX_CCR)—Offset 816Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 400h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RO	**Reserved (RSVD)**
19	0h RW	**USB3 Partition Engine/Link trunk gating Enable (PARUSB3_ENG_GEN):** When set to 1 enables gating of the SOSC trunk to the XHCI engine and link in the PARUSB3 partition.
18	0h RW	**USB3 Partition Frame Timer trunk gating Enable (PARUSB3_LINK_GEN):** When set to 1 enables gating of the SOSC trunk to the Frame timer in the PARUSB3 partition.
17	0h RW	**USB2 link partition clock gating enable (PARUSB2_CLK_GEN):** When set to 1 enables gating of the SOSC trunk to the USB2 link and Phy logic in the PARUSB2 partition.
16	0h RW	**USB2/USHIP 12.5 MHz partition clock gating enable (USHIP_PCGEN):** When set to 1 enables gating of the 12.5 MHz SOSC trunk to the USB2 and USHIP logic in the PARUSB2 partition.
15	0h RO	**Reserved1 (RSVD1):** Reserved
14	0h RW	**USB3 Port Aux/Core clock gating enable (USB3_AC_CGE):** When set, allows the aux_cclk clock into the USB3 port to be gated when conditions are met. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.
13:12	0h RW	**Rx Detect Timer when port Aux Clock is Gated (RX_DT_ACG):** This field defines the value of the timer used to perform Rx Detect when port Aux Clock has been gated. 0x0: 100ms 0x1: 12ms Others: Reserved Note: This timer shall use the Fast Training Timer Tick (about 1us tick) for simulation purposes. For Fast Training mode, the above timeouts will become about 11us and about 100us, +/- implementation uncertainty, respectively.
11:8	4h RW	**U2 Residency Before ModPHY Clock Gating (U2R_BM_CG):** Before gating ModPHY Aux clock, Host Controller shall wait for this time in U2. This time is meant to ensure that the attached device has entered U2 as well. 0x0: 1us 0x1: 128us 0x2: 256us 0x3: 512us 0x4: 640us 0x5: 768us 0x6: 896us 0x7: 1024us Others: Reserved Note: This counter shall start counting once pipe has entered PS3 state in response to link in U2.
7	0h RW	**Frame Timer Clock Gating Ports in U2 Enable (FTCGPU2E):** This bit, when set, allows Host Controller to gate the clock to the Frame Timer when ports are in U2.
6	0h RW	**USB2 port clock throttle enable (USB2_PC_TE):** When set, allows the Aux clock into the USB2 ports to be throttled when conditions allow.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**XHCI Engine Aux clock gating enable (XHCI_AC_GE):** When set, allows the aux clock into the XHCI engine to be gated when idle. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.
4	0h RW	**XHCI Aux PM block clock gating enable (XHCI_APMB_CGE):** When set, allows the aux clock into the Aux PM block to be gated when idle. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.
3	0h RW	**USB3 Aux Clock Trunk Gating Enable (USB3_AC_TGE):** When set, allows Aux Clock Trunk feeding to USB3.0 ports to be gated when port Aux clock is gated at all USB3.0 ports and all USB3.0 modPHY instances. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.
2	0h RW	**USB3 Port Aux/Port clock gating enable (USB3_AP_CGE):** When set, allows the aux_pclk clock into the USB3 port to be gated when conditions are met. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.
1	0h RW	**ModPHY port Aux clock gating enable in U2 (MPP_AC_GEU2):** When set, allows the aux clock into the ModPhy to be gated when Link is in U2 and pipe has been in PS3 for at least the time defined by U2 Residency Before ModPHY Clock Gating field. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.
0	0h RW	**ModPHY port Aux clock gating enable in Disconnected, U3 or Disabled (MPP_AC_GE_DDU3):** When set, allows the aux clock into the ModPHY to be gated when Link is in Disconnected, U3 or Disabled state. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.

16.258 USB LPM Parameters (USB_LPM_PARAM)—Offset 8170h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 96090032h

Bit Range	Default & Access	Field Name (ID): Description
31:22	258h RW	**USB2_LPM_REG_RSM_U3_DET_NORM (USB2_LPM_REG_RSM_U3_DET_NORM)**
21:19	1h RW	**Min U3 Exit LFPS Duration (MIN_U3E_LFPS_D):** Min U3 Exit LFPS Duration This field defines the minimum duration of LFPS driven by Host Controller upon U3 exit LFPS handshake. Note that theres an uncertainty of +-16us in actual duration driven by the Host Controller. 0b000: 96us 0b001: 160us 0b010: 224us 0b011: 288us 0b100: 352us 0b101: 416us 0b110: 480us 0b111: 544us
18:16	1h RW	**Min U2 Exit LFPS Duration (MIN_U2_ELFPS_D):** Min U2 Exit LFPS Duration This field defines the minimum duration of LFPS driven by Host Controller upon U2 exit LFPS handshake. Note that theres an uncertainty of +-16us in actual duration driven by the Host Controller. 0b000: 96us 0b001: 160us 0b010: 224us 0b011: 288us 0b100: 352us 0b101: 416us 0b110: 480us 0b111: 544us
15	0h RW	**Max PING LFPS Rx Detection (XHCI_MAX_PING_LFPS):** This field defines the maximum timing for PING LFPS. If an incoming LFPS will be considered a PING if it has a timing such that it is less than or equal to the selected value. Otherwise it will be considered for the other types of LFPS. 0b Max PING LFPS timing set to 256 ns (32 link clocks) 1b Max PING LFPS timing set to 320 ns (40 link clocks)
14:10	0h RO	**Reserved (RSVD_1)**
9:0	32h RW	**xHCI BESL to HIRD Distance (XHCI_BESL_HIRD_DT):** This field defines the gap between BESL and duration of Resume signalling from Host upon Host Initiated Resume from USB2.0 LPM. Default value of this register corresponds to xHCI spec defined 50us value. Value BESL to HIRD Distance 000h 0us 001h 1us 002h 2us 3FFh 1023us

16.259 xHC Latency Tolerance Parameters - LTV Control (XLTP_LTV1)—Offset 8174h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 40047Dh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Disable scheduler direct transition from IDLE to NO requirement (DIS_SDT_IDL_NR):** 0: (default) allow scheduler direct transition from IDLE to NO requirement 1: Disable scheduler direct transition from IDLE to NO requirement
30:26	0h RW	**Reserved (RSVD)**
25	0h RW	**XHCI LTR Transition Policy (XLTRTP):** When '0', LTR messaging state machine transitions from High, Medium, or Low LTR states to Active state upon the Alarm Timer timout and stays in Active until the next service boundary. When '1', the LTR messaging state machine transitions through High ? Med ? Low ? Active states assuming enough latency is available for each transition.
24	0h RW	**XHCI LTR Enable (XLTRE):** This bit must be set to enable LTV messaging from XHCI to the PMC.
23:12	400h RW	**Periodic Active LTV (PA_LTV):** 23:22 Latency Scale 00b : Reserved 01b : Latency Value to be multiplied by 1024 10b : Latency Value to be multiplied by 32,768 11b : Latency Value to be multiplied by 1,048,576 21:12 Latency Value (ns) Defaults to 0 micro seconds
11:0	47Dh RW	**USB2 Port L0 LTV (USB2_PL0_LTV):** 11:10 Latency Scale 00b : Reserved 01b : Latency Value to be multiplied by 1024 10b : Latency Value to be multiplied by 32,768 11b : Latency Value to be multiplied by 1,048,576 9:0 Latency Value (ns) Defaults to 128 Micro Seconds

16.260 xHC Latency Tolerance Parameters LTV Control 2 (XLTP_LTV2)—Offset 8178h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 17FFh

Bit Range	Default & Access	Field Name (ID): Description
31:13	0h RO	**Reserved (RSVD)**

Bit Range	Default & Access	Field Name (ID): Description
12:0	17FFh RW	**LTV Limit (LTV_LMT):** This register defines a maximum LTR value that is allowed to be advertised to the PMC. This is meant to be used as a workaround or mitigation if issues are discovered with the LTR values generated by the XHC using the defined algorithms. If the LTR value of the XHC is larger than the value in this register field, the value in this field is sent to the PMC instead. Default value is the highest possible - 101b 12:10: Latency Multiplier Field 000b - Value times 1 ns 001b - Value times 32 ns 010b - Value times 1,024 ns 011b - Value times 32,768 ns 100b - Value times 1,048,576 ns 101b - Value times 33,554,432 ns 110b-111b - Not Permitted 9:0: Latency Value Default = 3FFh

16.261 xHC Latency Tolerance Parameters - High Idle Time Control (XLTP_HITC)—Offset 817Ch

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**Reserved (RSVD)**
28:16	0h RW	**Minimum High Idle Time (MHIT):** This is the minimum schedule idle time that must be available before a "High" LTR value can be indicated. This value must be larger than HIWL 12:7 - Time value in # of 125 Micro Seconds Bus Intervals (0 - 8ms) 6:0 - Fractional BI Time value in Micro Seconds (0 - 124 Micro Seconds)
15:13	0h RO	**Reserved (RSVD_1)**
12:0	0h RW	**High Idle Wake Latency (HIWL):** This is the latency to access memory from the High Idle Latency state. This value must be larger than MIWL and LIWL 12:7 - Time value in # of 125 Micro Seconds Bus Intervals (0 - 8ms) 6:0 - Fractional BI Time value in Micro Seconds (0 - 124 Micro Seconds)

16.262 xHC Latency Tolerance Parameters - Medium Idle Time Control (XLTP_MITC)—Offset 8180h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**Reserved (RSVD)**
28:16	0h RW	**Minimum Medium Idle Time (MMIT):** This is the minimum schedule idle time that must be available before a "Medium" LTR value can be indicated. This value must be larger than MIWL 12:7 - Time value in # of 125 Micro Seconds Bus Intervals (0 - 8ms) 6:0 - Fractional BI Time value in Micro Seconds (0 - 124 Micro Seconds)
15:13	0h RO	**Reserved (RSVD_1)**
12:0	0h RW	**Medium Idle Wake Latency (MIWL):** This is the latency to access memory from the Medium Idle Latency state. This value must be larger than LIWL 12:7 - Time value in # of 125 Micro Seconds Bus Intervals (0 - 8ms) 6:0 - Fractional BI Time value in Micro Seconds (0 - 124 Micro Seconds)

16.263 xHC Latency Tolerance Parameters Low Idle Time Control (XLTP_LITC)—Offset 8184h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**Reserved (RSVD)**
28:16	0h RW	**Minimum Low Idle Time (MLIT):** This is the minimum schedule idle time that must be available before a "Low" LTR value can be indicated. This value must be larger than LIWL 12:7 - Time value in # of 125 Micro Seconds Bus Intervals (0 - 8ms) 6:0 - Fractional BI Time value in Micro Seconds (0 - 124 Micro Seconds)
15:13	0h RO	**Reserved (RSVD_1)**
12:0	0h RW	**Low Idle Wake Latency (LIWL):** This is the latency to access memory from the Medium Idle Latency state. 12:7 - Time value in # of 125 Micro Seconds Bus Intervals (0 - 8ms) 6:0 - Fractional BI Time value in Micro Seconds (0 - 124 Micro Seconds)

16.264 HOST_CTRL_BW_MAX3_REG (HOST_CTRL_BW_MAX3_REG)—Offset 8188h

Added for CHV

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: F42F42h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Rsvd (Rsvd)**
23:0	F42F42h RW	**MAX_OUT_BW_UNITS_FOR_SS_PORTS (MAX_OUT_BW_UNITS_FOR_SS_PORTS):** Max. Number of OUT BW units for SS ports - denominator in 90% calculation

16.265 PDDIS_REG (PDDIS_REG)—Offset 8198h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Rsvd (Rsvd)**
8:0	0h RW	**PDDISEN (PDDISEN)**

16.266 THRM_HOST_CTRL_REG (THRM_HOST_CTRL_REG)—Offset 819Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:21	0h RW	**Rsvd (Rsvd)**
20	0h RW	**SSIC Thermal Throttle Ux Mapping (SSIC_THRM_UX_MAPPING):** Controls if U1 or U2 is forced upon the start of thermal throttle OFF period. 0 Force ports into U2 during Thermal Throttle triggered Ux entry. 1 Force ports into U1 during Thermal Throttle triggered Ux entry.

Bit Range	Default & Access	Field Name (ID): Description
19:18	0h RW	**USB3 Thermal Throttle Ux LGO delay (USB3_THRM_UX_LGO_DELAY):** Controls the delay enforced between LMP for FLPMA and LMP for Ux LGO. After sending LPMA ON , wait for pre-defined number of clocks to initiate LGO_U1/ LGO_U2 00: 8 clocks 01: 32 clocks 10: 128 clocks 11: 0 clocks This field does not apply to ports that are not operating in the mode required to issue FLMA ON.
17	0h RW	**THRM_THROTTLING_DISABLE (THRM_THROTTLING_DISABLE):** 0: Thermal throttling is enabled. 1: Thermal throtling is disabled. The host controller ignores the TT control inputs and does not throttle.
16	0h RW	**USB3 Thermal Throttle Ux Mapping (USB3_THRM_UX_MAPPING):** Controls if U1 or U2 is forced upon the start of thermal throttle OFF period. 0 Force ports into U2 during Thermal Throttle triggered Ux entry. 1 Force ports into U1 during Thermal Throttle triggered Ux entry.
15	0h RW	**THROTTLE_PRIORITY_MODE (THROTTLE_PRIORITY_MODE):** 0: Off period has priority: In this case, when the throttle signal is asserted, the host controller enters the off state on the next uframe boundary, and stays in the off state for the prescribed duration or until the end of the 16 uframe throttle period whichever occurs first. On subsequent throttle periods, the off period occurs first and then the on period. 1: The On period has priority: In this case, when the throttle signal is asserted, the host controller completes the required On period first before entering the off period. If the required number of uFrames has already been executed in a 16 uframe throttle window, the controller enters the off period immediately.
14	0h RW	**DISABLE_FORCE_L1_WHEN_THROTTLED (DISABLE_FORCE_L1_WHEN_THROTTLED):** 0: USB2 port will not force L1 entry on throttled ports. L1 entry will be based on the normal idle timeout 1: USB2 ports will attempt to enter L1 immediately after throttled ports are idle.
13	0h RW	**DISABLE_INTERRUPT_THROTTLING (DISABLE_INTERRUPT_THROTTLING):** 0: Interrupt traffic is throttled 1: Interrupt traffic is not throttled
12	0h RW	**DISABLE_ISOCHRONOUS_THROTTLING (DISABLE_ISOCHRONOUS_THROTTLING):** 0: Isochronous traffic is throttled 1: Isochronous traffic is not throttled
11:8	0h RW	**T1_ACTION (T1_ACTION):** # bus intervals to be idle for async traffic out of the 16 interval master period; from 0 to 15.
7:4	0h RW	**T2_ACTION (T2_ACTION):** # bus intervals to be idle for async traffic out of the 16 interval master period; from 0 to 15.
3:0	0h RW	**T3_ACTION (T3_ACTION):** # bus intervals to be idle for async traffic out of the 16 interval master period; from 0 to 15.

16.267 LFPS_PM_CTRL_REG (LFPS_PM_CTRL_REG)—Offset 81A0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Rsvd (Rsvd)**
6:0	0h RW	**LFPS_PM_EN (LFPS_PM_EN)**

16.268 U2PDM (U2PDM)—Offset 81A4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Rsvd (Rsvd)**
8:0	0h RW	**U2PDM (U2PDM)**

16.269 U2PCM (U2PCM)—Offset 81A8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Rsvd (Rsvd)**
8:0	0h RW	**U2PCM (U2PCM)**

16.270 U3PDM (U3PDM)—Offset 81ACh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Rsvd (Rsvd)**
6:0	0h RW	**U3PDM (U3PDM)**

16.271 U3PCM (U3PCM)—Offset 81B0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Rsvd (Rsvd)**
6:0	0h RW	**U3PCM (U3PCM)**

16.272 THRM_HOST_CTRL_REG2 (THRM_HOST_CTRL_REG2)—Offset 81B4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 7Fh

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Rsvd (Rsvd)**
6:0	7Fh RW	**Force LPM Accept Enable (FORCE_LPM_ACCEPT_EN):** Per Port Control to allow for enforcement to be based on device detection if certain devices do not support FLPMA. 0: Do not set FLPMA prior to Ux entry due to TT 1: Set FLPMA prior to Ux entry due to TT

16.273 LFPSONCOUNT_REG (LFPSONCOUNT_REG)—Offset 81B8h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 20C8h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Rsvd (Rsvd)**
17:16	0h RW	**U2P3 LFPS Periodic Sampling Control (XU2P3LPSC):** This field controls the OFF time for the LFPS periodic sampling for SS and SSIC ports in U2P3. If LFPSPM for a port is 1, it will override the OFF time and LFPS receiver will remain OFF permanently. For Fast Sim mode, 500us will be equivalent to 5us. 0x0 Polling Disable. (RXDET Polling will become 100ms.) 0x1 500us OFF Time 0x2 1ms OFF Time 0x3 1.5ms OFF Time
15:10	8h RW	**XLFPSONCNTSSIC (XLFPSONCNTSSIC):** This time would describe the number of clocks SSIC LFPS will remain ON. SSIC LFPS detection operation may be carried out on using RTC clock or Oscillator clock. The value of this register should be adjusted accordingly. For RTC recommended value is 2. For Oscillator clock, recommended value is 8.
9:0	C8h RW	**XLFPSONCNTSS (XLFPSONCNTSS):** This time would describe the number of clocks LFPS will remain ON. LFPS detection operation may be carried out on using RTC clock or Oscillator clock. The value of this register should be adjusted accordingly. For RTC recommended value is 2. For Oscillator clock, recommended value is 200.

16.274 (USB2PMCTRL_REG)—Offset 81C4h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD)**
11	0h RW	**USB2 PHY SUS Power Gate PORTSC Block Policy (U2PSPGPSCBP):** This controls the policy for blocking PORTSC Updates while the USB2 PHY SUS Well is power gated. When set, the controller will block any updates to the PORTSC caused by port status change if the USB2 PHY SUS is power gated. 0 Do not

Bit Range	Default & Access	Field Name (ID): Description
10:8	0h RW	**USB2 PHY SUS Well Power Gate Entry Hysteresis Count (U2PSPGEHC):** This controls the amount of hysteresis time the controller will enforce after detecting the USB2 PHY SUS Power Gate entry condition. 0h 0 clocks 1h 32 clocks 2h 64 clocks 3h 128 clocks 4h 256 clocks 5h 512 clocks 6h 1024 clocks 7h 2048 clocks
7:4	0h RW	**USB2 PHY SUS Power Gate PORTSC Block Policy (U2CLPGLAT):** This field represents the worst case latency for the USB2 Common Lane to enter and exit its power gate state. This fields is required to be compared to a ports HIRD/HIRD value for the ports that have allowed L1 to L2 mapping to determine if the Common Lane can be allowed to power off. If the power gate entry/ exit latency is greater than the HIRD/HIRDD then the common lane should not be allowed to power gate as this will result in a L1 exit violation. 0h 100 us 1h 200 us 2h 300 us Eh 1500us Fh 1600 us
3:2	0h RW	**USB2 PHY SUS Well Power Gate Policy (U2PSUSPGP):** This field controls when to enable the USB2 PHY SUS Well Power Gating when the proper conditions are met. 00 USB2 PHY SUS Power Gating is Disabled. 01 USB2 PHY SUS Power Gating is Enabled in Only D0 (Excludes D0i3 and D3) 10 USB2 PHY SUS Power Gating is Eanabled in only in D0 and D0i3 (Excludes D3) 11 USB2 PHY SUS Power Gating is Eanabled in D0/D0i3/D3
1	0h RW	**USB2 Common Lane Power Gating Enable During L1 to L2 Mapping for USB2 PHY Power Gating (U2CLPGEL1L2):** This field when set enables the controller to allow for the common lane power gating to be enabled when all ports are exposed as in L2 to the USB2 PHY while at least 1 port has been mapped to L2 from L1. This field alone does not guarantee power gating since the L1 HIRD/HIRDD Value must be compared with the PHYs power gate exit latency (U2CLPGLAT) held in this register to ensure that L1 exit is not violated. 0 USB2 Common Lane Power Gating is disabled when any port has been mapped from L1 to L2. 1 USB2 Common Lane Power Gating is allowed when any port has been mapped to L2 from L1 with the additional condition that the HIRD/HIRDD is greater than the PHYs Power Gate exit latency.
0	0h RW	**USB2 Data Lane L1 to L2 Mapping Enable for USB2 PHY Power Gating (U2DLL1L2ME):** This field when set enables the controller to map an L1 entry directly to L2 to allow the USB2 PHY to trigger its Autonomous Power Gating. The USB2 PHY will trigger PG only when in L2 since it does not fully understand the requirements for L1. 0 USB2 L1 to L2 mapping is disabled for all ports 1 USB2 L1 to L2 mapping is enabled for all ports

16.275 ECC_PARITY_ERROR_LOG_REG (ECC_PARITY_ERROR_LOG_REG)—Offset 83F8h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C	**COMMAND_PARITY_DETECTED (COMMAND_PARITY_DETECTED):** Command Parity Error detected on received IOSF transaction
30	0h RW/1C	**DATA_PARITY_DETECTED (DATA_PARITY_DETECTED):** Data Parity Error detected on received IOSF transaction
29	0h RW/1C	**ERROR_PRESENT_DETECTED (ERROR_PRESENT_DETECTED):** Error present detected on received IOSF transaction
28:26	0h RO	**RSVD (RSVD):** Reserved
25:21	0h RW/1C	**Correctable ECC Error Source RF (CORRECTABLE_ECC_ERROR_SOURCE_RF):** When set indicates the specific RF the ECC Error was detected on. This is used along with the ECC Source, and Port Info to determine the specific RF on which Correctable ECC Error was seen on. USB2 Port RF: 00000: Reserved 00001: Async. TX Payload 00010: Periodic TX Payload 00011: RX Payload 00100: TTE Periodic TX Payload 00101: TTE RX Payload Others: Reserved USB3 Port RF: 00000: Reserved 00001: Async TX Payload 00010: Periodic TX Payload 00011: RX Payload Others: Reserved XHCI Engine RF: 00000: Reserved 00001: TTE Context 00010: TRM Context 00011: XHCI Completion Collect 00100: Debug Device Completion Collect 00101: DBC-EXI Trace Data
20:15	0h RW/1C	**CORRECTABLE_ECC_ERROR_SOURCE_PORT (CORRECTABLE_ECC_ERROR_SOURCE_PORT):** When set indicates the Port# if the ECC Error was detected on any of the USB2/USB3 Port RFs. This is valid only when bit[1:0] != 00
14:13	0h RW/1C	**CORRECTABLE_ECC_ERROR_SOURCE_LOG (CORRECTABLE_ECC_ERROR_SOURCE_LOG):** When an Correctable ECC is detected for an RF, the corresponding bit is set to '1' 11: Detected Uncorrectable ECC Error on USB3 RFs 10 : Detected Uncorrectable ECC Error on USB2 RFs 01: Detected Uncorrectable ECC Error on XHCI Engine RFs 00 : No Error Note: On detection of first Uncorrectable ECC Error HW shall lock the Correctable log registers (Source, Port, RF) until SW clears the ECC Error log or cleared by reset.

Bit Range	Default & Access	Field Name (ID): Description
12:8	0h RW/1C	**UNCORRECTABLE_ECC_ERROR_SOURCE_RF (UNCORRECTABLE_ECC_ERROR_SOURCE_RF):** When set indicates the specific RF the ECC Error was detected on. This is used along with the ECC Source, and Port Info to determine the specific RF on which uncorrectable ECC Error was seen on. USB2 Port RF: 00000: Reserved 00001: Async. TX Payload 00010: Periodic TX Payload 00011: RX Payload 00100: TTE Periodic TX Payload 00101: TTE RX Payload Others: Reserved USB3 Port RF: 00000: Reserved 00001: Async TX Payload 00010: Periodic TX Payload 00011: RX Payload Others: Reserved XHCI Engine RF: 00000: Reserved 00001: TTE Context 00010: TRM Context 00011: XHCI Completion Collect 00100: Debug Device Completion Collect 00101: DBC-EXI Trace Data
7:2	0h RW/1C	**UNCORRECTABLE_ECC_ERROR_SOURCE_PORT (UNCORRECTABLE_ECC_ERROR_SOURCE_PORT):** When set indicates the Port# if the ECC Error was detected on any of the USB2/USB3 Port RFs. This is valid only when bit[1:0] != 00
1:0	0h RW/1C	**UNCORRECTABLE_ECC_ERROR_SOURCE_LOG (UNCORRECTABLE_ECC_ERROR_SOURCE_LOG):** When an uncorrectable ECC is detected for an RF, the corresponding bit is set to '1' 11: Detected Uncorrectable ECC Error on USB3 RFs 10 : Detected Uncorrectable ECC Error on USB2 RFs 01: Detected Uncorrectable ECC Error on XHCI Engine RFs 00 : No Error Note: On detection of first Uncorrectable ECC Error HW shall lock the Uncorrectable log registers (Source, Port, RF) until SW clears the ECC Error log or cleared by reset.

16.276 ECC_POISONING_CTRL_REG (ECC_POISONING_CTRL_REG)—Offset 83FCh

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RW	**ENGINE_RF_ECC_POISONING_VECTOR (ENGINE_RF_ECC_POISONING_VECTOR):** XHCI Engine RFs ECC Poisoning Inversion bits for the 64b or 128bit based on RF width
22:14	0h RW	**USB3_RF_ECC_POISONING_VECTOR (USB3_RF_ECC_POISONING_VECTOR):** USB3 RFs ECC Poisoning Inversion bits for the 64b or 128bit based on RF width

Bit Range	Default & Access	Field Name (ID): Description
13:5	0h RW	**USB2_RF_ECC_POISONING_VECTOR (USB2_RF_ECC_POISONING_VECTOR):** USB2 RFs ECC Poisoning Inversion bits for the 64b or 128bit based on RF width
4:3	0h RO	**RSVD (RSVD)**
2	0h RW	**ENGINE_RF_ECC_POISONING_EN (ENGINE_RF_ECC_POISONING_EN):** Enable ECC Poisoning for XHCI Engine related RFs that support ECC
1	0h RW	**USB3_RF_ECC_POISONING_EN (USB3_RF_ECC_POISONING_EN):** Enable ECC Poisoning for USB3 Port related RFs that support ECC. Setting this bit enables poisoning of USB3 Port related RFs. This applies to all USB3 ports
0	0h RW	**USB2_RF_ECC_POISONING_EN (USB2_RF_ECC_POISONING_EN):** Enable ECC Poisoning for USB2 Port related RFs that support ECC. Setting this bit enables poisoning of USB2 Port related RFs. This applies to all USB2 ports

16.277 USB2_PORT_STATE_REG (USB2_PORT_STATE_REG)—Offset 8400h

Access Method

Type: MEM Register (Size: 64 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:18	0h RO	**RSVD (RSVD)**
17:0	0h RO	**USB2_PORT_STATE_REG (USB2_PORT_STATE_REG):** Per USB2 Port State Register indicating the following states 0x0 - Connected 0x1 - Suspended 0x2 - Disabled 0x3 - Reserved This register is Read Only and will reflect the state of the ports. Status can change at any time and any usage of this information must be done with care and under specific flows where port state change race conditions are properly handled.

16.278 USB3_PORT_STATE_REG (USB3_PORT_STATE_REG)—Offset 8408h

Access Method

Type: MEM Register (Size: 64 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:14	0h RO	**RSVD (RSVD)**
13:0	0h RO	**USB3_PORT_STATE_REG (USB3_PORT_STATE_REG):** Per USB3 Port State Register indicating the following states 0x0 - Connected 0x1 - Suspended 0x2 - Disabled 0x3 - Reserved This register is Read Only and will reflect the state of the ports. Status can change at any time and any usage of this information must be done with care and under specific flows where port state change race conditions are properly handled.

16.279 FUS1_REG (FUS1_REG)—Offset 8410h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 10000h

Bit Range	Default & Access	Field Name (ID): Description
31:21	0h RO	**RSVD (RSVD)**
20:18	0h RO	**DEV_ID (DEV_ID):** Provides a fuse over-ride for the lower 3 bits of device ID.
17	0h RO	**XHC_DPGDIS (XHC_DPGDIS):** When asserted, it indicates that the xHCI will not dynamically power gate the controller.
16	1h RO	**USBR_DIS (USBR_DIS):** When asserted, it indicates that xHCI does not support USBr
15	0h RO	**XHC_FXN_DIS (XHC_FXN_DIS):** When asserted, it indicates the xHCI is fused to function disabled.
14:10	0h RO	**USB2_PORT_COUNT (USB2_PORT_COUNT):** 0x0 = MAX USB2 Ports enabled 0x1 = Max USB2 Ports -1 enabled 0x1F = All ports disabled
9:5	0h RO	**USB3_PORT_COUNT (USB3_PORT_COUNT):** 0x0 = MAX USB3 Ports enabled 0x1 = Max USB3 Ports -1 enabled 0x1F = All ports disabled
4	0h RO	**DEBUG_MODE_ENABLE (DEBUG_MODE_ENABLE):** 0 = USB debug mode disabled 1 = USB debug mode enabled Not used for LPT.
3	0h RO	**XHC_DCGDIS (XHC_DCGDIS):** 0= USB3 (xHC) dynamic clock gating enabled 1= USB3 (xHC) dynamic clock gating disabled
2	0h RO	**REGFILE_DPGDIS (REGFILE_DPGDIS):** 0 = USB3 (xHC) dynamic RF power gating enabled 1 = USB3 (xHC) dynamic RF power gating disabled

Bit Range	Default & Access	Field Name (ID): Description
1	0h RO	**USBIO_PMDIS (USBIO_PMDIS):** 0=USB2 HW LPM and USB3 HW Ux under xHC enabled 1= USB2 HW LPM and USB3 HW Ux under xHC disabled
0	0h RO	**USB2_PLL_SHUTDOWN_DIS (USB2_PLL_SHUTDOWN_DIS):** 0= USB2 PLL Shutdown enabled 1= USB2 PLL Shutdown disabled

16.280 FUS2_REG (FUS2_REG)—Offset 8414h

This register is NOT subject to HW save and restore.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**RSVD (RSVD)**
11:0	0h RO	**PORT_MAP_FUS1 (PORT_MAP_FUS1):** 1:0 Port 1 3:2 Port 2 5:4 Port 3 31:30 Port 16 Bit Description: 00: Port assgned to USB3 01: Port assigned to non-XHCI controller 10: Port assigned to Flex IO mapping. Mapping is based on Soft Straps. 11: Reserved.

16.281 FUS3_REG (FUS3_REG)—Offset 8418h

This register is NOT subject to HW save and restore.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RSVD (RSVD)**

16.282 STRAP1_REG (STRAP1_REG)—Offset 841Ch

This register is NOT subject to HW save and restore.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RO	**RSVD (RSVD)**
5:0	0h RO	**FlexIO mapping (FLEX_IO_MAPPING):** Each bit corresponds to the appropriately numbered USB3 port (zero based) Bit 0 = port 1, bit 1 = port 2 etc. 0: Port is owned by XHCI 1: Port is not owned by XHCI

16.283 STRAP2_REG (STRAP2_REG)—Offset 8420h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RO	**RSVD (RSVD)**
5:0	0h RO	**USB3_SSIC_MODE (USB3_SSIC_MODE):** Each bit corresponds to the appropriately numbered USB3 port (zero based) Bit 0 = port 1, bit 1 = port 2 etc. 0: Port is operated in USB3 mode 1: Port is operated in SSIC mode.

16.284 STRAP3_REG (STRAP3_REG)—Offset 8424h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**RSVD (RSVD)**
1:0	0h RO	**XHCI_PRI_CLK_FREQ_SEL (XHCI_PRI_CLK_FREQ_SEL):** Specifies the frequency of the Primary clock (iosf_prim_mux_clk) used by XHCI 00 : 200 MHz (default) 01 : 125 MHz 10 : 250 MHz

16.285 DFT_REG1 (DFT_REG1)—Offset 8430h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**SS/SSIC DFT CRC (SSDFTCRC):** These register bits contain the value of SS DFT CRC
15:12	0h RO	**RSVD0 (RSVD0):** RSVD0
11	0h RW	**DFTLFPSSEL (DFTLFPSSEL):** DFTLFPSSEL 0: Rxelecidle is driven from GPIO Pin 1: Rxelecidle is Driven as specified by Super Speed DFT LFPS Mode Select
10	0h RW	**Super Speed DFT LFPS Mode Select (SSDFTLMSEL):** This bit selects the Super Speed DFT LFPS mode, and will only take effect when Super Speed HBP mode is enabled. 0b: HBP logic will internally loopback TX LFPS as RX LFPS and AFE RX LFPS path to the controller is disconnected. 1b: RX LFPS path works normally
9:6	0h RW	**SS/SSIC DFT CRC Select (SSDFTCRCSEL):** These bits select which Super Speed DFT CRC value is reflected in SSDFTCRC bits. In addition, these bits also select which SuperSpeed DFT CRC MSB value is sent to GPIO monitor pin, i.e. sata3gp_gp37 to aid silicon debug. Live version of selected CRC's MSB will be- sent to GPIO monitor pin, i.e. sata3gp_gp37. 0h (default): No SuperSpeed DFT CRC is selected. 1h: Data Payload CRC 2h: Link Management Packet CRC 3h: Transaction Packet CRC 4h: Isochronous Timestamp Packet CRC 5h: Data Packet Header CRC 6h: Link Command Packet CRC 7h: RRAP Packet CRC 8h: Tx/Rx Cfg CRC Others: Reserved Note : CRC types 7 and 8 are onlyapplicable if Port is SSIC Port
5	0h RO	**RSVD1 (RSVD1):** RSVD1
4:0	0h RW	**SS/SSIC DFT CRC Port Select (SSICDFTCPS):** One CRC per packet type is shared for all the Super Speed ports. These bits select the Super Speed port for which CRC data will be updated. 000b: (default) No SuperSpeed Port is selected 001b: SS/SSIC Port 0 010b: SS/SSIC Port 1 011b: SS/SSIC Port 2 100b: SS/SSIC Port 3 101b: SS/SSIC PORT4 110b :SS/SSIC Port 5 others : Rsvd

16.286 DFT_REG2 (DFT_REG2)—Offset 8434h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**HS/HSIC TX CRC (TXCRC):** These register bits contain the value of TX CRC. This TX CRC is placed right after the 480MHz XCLKQ.
15:11	0h RW	**HSIC/UTMI+ DFT Port Select (UTMIDFTPS):** One CRC is shared for all the UTMI+ ports. These bits select the UTMI+ port for which CRC data will be updated and loopback status reflected in UTMILPBKSTS. 0h: (default) No UTMI+ Port is selected 1h: UTMI+ Port 0 2h: UTMI+ Port 1 3h: UTMI+ Port 2 4h: UTMI+ Port 3 5h: UTMI+ Port 4 6h: UTMI+ Port 5 7h: UTMI+ Port 6 8h: UTMI+ Port 7 9h: UTMI+ Port 8 Ah: UTMI+ Port 9 Bh: UTMI+ Port 10 Ch: UTMI+ Port 11 Dh: UTMI+ Port 12 Eh: UTMI+ Port 13 Others: Reserved
10:7	0h RW	**Loop Number (UTMILPBKLOOPN_3_0):** Number of repeatable fixed pattern within a packet Note: Connect register bit 3 to counter bit 7, register bit 2 to counter bit 5, register bit 1 to counter bit 3, register bit 0 to counter bit 1. Counter bits 6, 4, 2 and 0 will be tied off to 0. Hence, the programmable loop number shall be: 0000b: 0 loop 0001b: 2 loops 0010b: 8 loops 0011b: 10 loops 1100b: 160 loops 1101b: 162 loops 1110b: 168 loops 1111b: 170 loops (max) +E27
6:5	0h RW	**Operational Mode (UTMIOPMODE_1_0):** Operational Mode in test mode. These signals select between various operational modes: 00b: Normal Operation 01b: Non-Driving 10b: Disable Bit Stuffing and NRZI encoding 11b: Reserved
4	0h RW	**Termination Select (UTMITERMSEL):** Termination Select in test mode. This signal selects between the FS and HS terminations: 0b: HS termination enabled 1b: FS termination enabled
3:2	0h RW	**Transceiver Select (UTMIXCVRSELECT_1_0):** Transceiver Select in test mode. This signal selects between the LS, FS and HS transceivers: 00b: HS transceiver enabled 01b: FS transceiver enabled 10b: LS transceiver enabled 11b: Reserved
1:0	0h RW	**UTMI+ Loopback Status (UTMILPBKSTS):** Loopback Status for port selected by UTMIDFTPS 00b: Reset condition 01b: Comparator has started receiving data and the received data matches with the TX pattern. 10b: Comparator has started receiving data the received data does not match with the TX pattern but there was no assertion of RC error from UTMI. 11b: Comparator has started receiving data and RX ERROR was asserted for at least one clock by UTMI. Note that this does not reflect the status of pattern comparison since RX error from UTMI is unexpected for loopback.

16.287 DFT_REG3 (DFT_REG3)—Offset 8438h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Loopback Pattern (UTMILPBKPAT):** 2-byte pattern for loopback Note: This 2-byte pattern will be replicated to the upper 2-byte to form the DW pattern
15:2	0h RW	**Loopback Lane Select (UTMILPBKSEL_13_0):** Port is selected if the corresponding bit is selected Note: MSB is for UTMI+ Port13, LSB is for UTMI+ Port0
1	0h RW	**Loopback Pattern (UTMILPBKPATSEL):** Near end loopback pattern generation type: 0b: Fixed pattern 1b: USB2 test packet
0	0h RW	**Loopback Type (UTMILPBKTYPE):** Loopback Type in test mode. This signal selects between DNELB and ANELB, and will only take effect if DTUTMILPBKEN is set. 0b: Digital Near-End Loopback (DNELB) 1b: Analog Near-End Loopback (ANELB) Note: Analog Far-End Loopback (AFELB) is not supported

16.288 dft_reg4 (DFT_REG4)—Offset 843Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RW	**RSVD (RSVD)**
10:0	0h RW	**Loopback Pattern (UTMILPBKPATSEL):** Loopback Lane Select (UTMILPBKSEL): Port is selected if the corresponding bit (zero based) is selected Bit 0 = Port 1 Bit 1 = Port 2 Etc.

16.289 dft_reg5 (DFT_REG5)—Offset 8440h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW/L	**High Speed Bypass Enable (HIGH_SPEED_BYPASS_ENABLE):** 0000 - Disable all HBP 0001 - Enable SS HBP 0010 - Enable HS HBP 0011 - Enable HSIC HBP 0100 - Enable SSIC HBP Others - Reserved

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**DFTIDPINSEL (DFTIDPINSEL):** Select Between Device and Host Controller in HBP mode 0h : Device Controller is selected 1h : Host Controller is selected
26	0h RW/O	**Lock Bit (LOCK):** This bit is used by BIOS to lock/unlock lockable bits. When set to '1' the write access to HBP Enable is disabled (locked state). When set to '0', the write access to bit locked are enable (unlocked state). Writable once after platform reset.
25	0h RW/L	**Loopback Enable (UTMILPBKEN):** Enable loopback test mode. If asserted, loopback test mode is enabled
24:0	0h RO	**Rsvd (Rsvd)**

16.290 XECP_CMDM_STS0 (XECP_CMDM_STS0)—Offset 8448h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**IDMA_OWNS_CNTX (IDMA_OWNS_CNTX):** Indicates that IDMA module owns the context access currently
30	0h RO	**ODMA_OWNS_CNTX (ODMA_OWNS_CNTX):** Indicates that ODMA module owns the context access currently
29	0h RO	**TRM_OWNS_CNTX (TRM_OWNS_CNTX):** Indicates that TRM modules owns the context access currently
28	0h RO	**CMD_RING_REQUESTED_CNTX_LOCK (CMD_RING_REQUESTED_CNTX_LOCK):** Indicates that Command Manager has requested a context lock
27	0h RO	**CMD_RING_STOP_IN_PROGRESS (CMD_RING_STOP_IN_PROGRESS):** Indicates that Command Ring stop command is in progress
26	0h RO	**SCH_UPDATE_CLR_EP_IN_PROGRESS (SCH_UPDATE_CLR_EP_IN_PROGRESS):** Indicates that clearing an EP out of schedule is in progress
25	0h RO	**ADDR_DEV_DONE (ADDR_DEV_DONE):** Indicates that current address device command is done by ODMA
24	0h RO	**ADDR_DEV_IN_PROGRESS (ADDR_DEV_IN_PROGRESS):** Indicates that ODMA has an address device command in progress
23	0h RO	**EP_STATE_UPDATE_IN_PROGRESS (EP_STATE_UPDATE_IN_PROGRESS):** Indicates that updating of EP state is in progress

Bit Range	Default & Access	Field Name (ID): Description
22	0h RO	**EP_STATE_IN_PROGRESS_FROM_STOP_DUE_TO_DB (EP_STATE_IN_PROGRESS_FROM_STOP_DUE_TO_DB):** Indicates that doorbell manager is issuing and EP update due to a doorbell ring on an EP that is in stop state
21	0h RO	**EP_STATE_UPDATE_IN_PROGRESS_DUE_TO_EP_ERROR (EP_STATE_UPDATE_IN_PROGRESS_DUE_TO_EP_ERROR):** Indicates that transfer ring manager is isuing and EP update due to an EP error condition detected
20	0h RO	**EP_STATE_UPDATE_IN_PROGRESS_DUE_TO_STALL (EP_STATE_UPDATE_IN_PROGRESS_DUE_TO_STALL):** Indicates that transfere ring manager is issuing an EP state update due to stall received
19	0h RO	**Reserved (RSVD)**
18	0h RO	**STOP in progress (STOP_IN_PROGRESS):** Indicates that a STOP on the Command Ring is in progress
17	0h RO	**commend ring has doorbell pending (CMD_RING_DB_PENDING):** Indicates that the commend ring has doorbell pending
16	0h RO	**commend ring running (CMD_RING_RUNNING):** Indicates that the commend ring is running
15:8	0h RO	**Command next capability offset (CMD_NEXT_CAP_OFFSET)**
7:0	0h RO	**Vendor defined capability ID (VID)**

16.291 XECP_CMDM_STS1 (XECP_CMDM_STS1)—Offset 844Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 3FC0000h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Rsvd0 (Rsvd0)**
25:18	FFh RO	**Event manager Producer Cycle State (EMPCS)**
17:12	0h RO	**Interrupter 7 TRB Count [5:0] (INT7_TRB_CNT)**
11:6	0h RO	**Interrupter 6 TRB Count [5:0] (INT6_TRB_CNT)**
5:0	0h RO	**Interrupter 5 TRB Count [5:0] (INT5_TRB_CNT)**

16.292 XECP_CMDM_STS2 (XECP_CMDM_STS2)—Offset 8450h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RO	**Rsvd0 (Rsvd0)**
4:0	0h RO	**Event Ring Segment Table (ERST):** count low

16.293 XECP_CMDM_STS3 (XECP_CMDM_STS3)—Offset 8454h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd0 (Rsvd0)**
15:0	0h RO	**Event Ring Segment Table (ERST):** count high

16.294 XECP_CMDM_STS4 (XECP_CMDM_STS4)—Offset 8458h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Event Ring Enqueue Pointer Low (EREPL)**

16.295 XECP_CMDM_STS5 (XECP_CMDM_STS5)—Offset 845Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Event Ring Enqueue Pointer High (EREPH)**

16.296 AUX Power PHY Reset (UPORTS_PON_RST_REG)—Offset 8460h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**RESERVED (RSVD)**
3:0	0h WO	**Allow Software USB PHY RST (ALL_SW_UP_RST):** Allow a USB PHY reset being issued by software. Writing to this register with bit set to 1 will reset the USB PHY that is connected to the port. Bit3:0 indicates the port number of the USB PHY

16.297 Latency Tolerance Control 0 (HOST_IF_LAT_TOL_CTRL_REG0)—Offset 8464h

The Latency Tolerance Control Register is used by SW to control which BELT is returned when this register is read. SW shall write to this register to program a Slot-ID, Port-ID and BELT Select to determine which BELT is selected. When this register is read the selected BELT is returned.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: D0000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h WO	**BELT Select (BELT_SEL):** This field determines what value will be selected to read back from SW when reading this register 0: Returns the SW programmed Latency Tolerance Value 1: Returns the Lowest BELT in the Host 2: Returns the BELT for the requested Slot-ID (Slot Select) 3: Returns the BELT for the requested Port-ID (Port Select)
29:20	0h RO	**Rsvd1 (Rsvd1)**
19:16	Dh WO	**Port Select (PORT_SEL):** Used to select the BELT for a given Port # when the BELT Select is programmed to select the Port-ID (this field is 0 based)
15:12	0h RO	**Rsvd (Rsvd)**
11:5	0h RO	**BELT Value (BELTV):** Value of selected BELT is return in this field
4:0	0h RW	**Slot Select (SLOT_SEL):** Reads will return: BELT Value (BELTV) [4:0]: Value of selected BELT is return in this field Writes will control : Slot Select (): Used to select the BELT for a given Slot # when the BELT Select is programmed to select the Slot-ID (this field is zero based)

16.298 USB Legacy Support Capability (USBLEGSUP)—Offset 846Ch

This register is modified and maintained by BIOS

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 2201h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Rsvd2 (Rsvd2)**
24	0h RW	**HC OS Owned Semaphore (HCOSOS)**
23:17	0h RO	**Rsvd1 (Rsvd1)**
16	0h RW	**HC BIOS Owned Semaphore (HCBIOSOS)**
15:8	22h RW/S	**Next Capability Pointer (NextCP)**
7:0	1h RW/L	**Capability ID (CID)**

16.299 USB Legacy Support Control Status (USBLEGCTLSTS)—Offset 8470h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/C	**SMI on BAR (SMIBAR)**
30	0h RW/C	**SMI on PCI Command (SMIPCIC)**
29	0h RW/C	**SMI on OS Ownership Change (SMIOSOC)**
28:21	0h RO	**Rsvd4 (Rsvd4)**
20	0h RO	**SMI on Host System Error (SMIHSE)**
19:17	0h RO	**Rsvd3 (Rsvd3)**
16	0h RO	**SMI on Event Interrupt (SMIEI)**
15	0h RW	**SMI on BAR Enable (SMIBARE)**
14	0h RW	**SMI on PCI Command Enable (SMIPCICE)**
13	0h RW	**SMI on OS Ownership Enable (SMIOSOE)**
12:5	0h RO	**Rsvd2 (Rsvd2)**
4	0h RW	**SMI on Host System Error Enable (SMIHSEE)**
3:1	0h RO	**Rsvd1 (Rsvd1)**
0	0h RW	**USB SMI Enable (USBSMIE)**

16.300 Port Disable Override capability register (PDO_CAPABILITY)—Offset 84F4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 3C6h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd (Rsvd)**
15:8	3h RO	**Next Capability Pointer (NCP)**
7:0	C6h RO	**Capability ID (CID)**

16.301 USB2 Port Disable Override (USB2PDO)—Offset 84F8h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Rsvd1 (Rsvd1)**
8:0	0h RW/O	**USB2PDO (USB2PDO)**

16.302 USB3 Port Disable Override (USB3PDO)—Offset 84FCh

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Rsvd1 (Rsvd1)**
6:0	0h RW/O	**USB3 Port Disable Override (USB3PDO):** A '1' in a bit position prevents the corresponding USB3 port from reporting a Device Connection to the XHC.

16.303 HW state capability register (HW_STATE_CAPABILITY)—Offset 8500h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 1F40C7h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Rsvd (Rsvd)**
23:16	1Fh RO	**Save Length (SAVE_LENGTH):** Indicates the number of DWords in this capability starting at offset 04h, that need to be saved and restored
15:8	40h RO	**Next Capability Pointer (NCP)**
7:0	C7h RO	**Capability ID (CID)**

16.304 HW state register 1 (HW_STATE_REG1)—Offset 8504h

Implementation defined HW state: SW must not access this register.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**HW_STATE_REG1 (HW_STATE_REG1)**

16.305 HW state register 2 (HW_STATE_REG2)—Offset 8508h

Implementation defined HW state: SW must not access this register.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**HW_STATE_REG2 (HW_STATE_REG2)**

16.306 HW state register 3 (HW_STATE_REG3)—Offset 850Ch

Implementation defined HW state: SW must not access this register.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**HW_STATE_REG3 (HW_STATE_REG3)**

16.307 HW state register 4 (HW_STATE_REG4)—Offset 8510h

Implementation defined HW state: SW must not access this register.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**HW_STATE_REG4 (HW_STATE_REG4)**

16.308 CONFIG mirror capability register (CONFIG_MIRROR_CAPABILITY)—Offset 8600h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 40C2h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd (Rsvd)**
15:8	40h RO	**Next Capability Pointer (NCP)**
7:0	C2h RO	**Capability ID (CID)**

16.309 Command (CMD_MMIO)—Offset 8604h

Dummy register, mirror of physical register as CMD

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:11	0h RO	**Reserved (RSVD)**
10	0h RW	**Interrupt Disable (ID):** When cleared to 0, the function is capable of generating interrupts. When 1, the function can not generate its interrupt to the interrupt controller. Note that the corresponding Interrupt Status bit is not affected by the interrupt enable.
9	0h RO	**Fast Back to Back Enable (FBE)**
8	0h RW	**SERR# Enable (SERR):** When set to 1, the XHC is capable of generating (internally) SERR#.
7	0h RO	**Wait Cycle Control (WCC)**
6	0h RW	**Parity Error Response (PER):** When set to 1, the XHCI Host Controller will check for correct parity (on its internal interface) and halt operation when bad parity is detected during the data phase as recommended by the XHCI specification. Note that this applies to both requests and completions from the system interface. This bit must be set in order for the parity errors to generate SERR#.
5	0h RO	**VGA Palette Snoop (VPS)**
4	0h RO	**Memory Write Invalidate (MWI)**
3	0h RO	**Special Cycle Enable (SCE)**
2	0h RW	**Bus Master Enable (BME):** When set, it allows XHC to act as a bus master. When cleared, it disable XHC from initiating transactions on the system bus.
1	0h RW	**Memory Space Enable (MSE):** This bit controls access to the XHC Memory Space registers. If this bit is set, accesses to the XHC registers are enabled. The Base Address register for the XHC should be programmed before this bit is set.
0	0h RO	**I/O Space Enable (IOSE):** Reserved as 0. Read-Only.

16.310 Device Status (STS_MMIO)—Offset 8606h

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 290h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**Detected Parity Error (DPE):** This bit is set by the Intel PCH whenever a parity error is seen on the internal interface to the XHC host controller, regardless of the setting of bit 6 or bit 8 in the Command register or any other conditions. Software clears this bit by writing a 1 to this bit location.
14	0h RW/1C	**Signaled System Error (SSE):** This bit is set by the Intel PCH whenever it signals SERR# (internally). The SERR_EN bit (bit 8 in the Command Register) must be 1 for this bit to be set. Software clears this bit by writing a 1 to this bit location.
13	0h RW/1C	**Received Master-Abort Status (RMA):** This bit is set when XHC, as a master, receives a master-abort status on a memory access. This is treated as a Host Error and halts the DMA engines. Software clears this bit by writing a 1 to this bit location.
12	0h RW/1C	**Received Target Abort Status (RTA):** This bit is set when XHC, as a master, receives a target abort status on a memory access. This is treated as a Host Error and halts the DMA engines. Software clears this bit by writing a 1 to this bit location.
11	0h RW/1C	**Signaled Target-Abort Status (STA):** This bit is used to indicate when the XHC function responds to a cycle with a target abort.
10:9	1h RO	**DEVSEL# Timing Status (DEVT):** This 2-bit field defines the timing for DEVSEL# assertion. Read-Only.
8	0h RW/1C	**Master Data Parity Error Detected (MDPED):** This bit is set by the Intel PCH whenever a data parity error is detected on a XHC read completion packet on the internal interface to the XHC host controller and bit 6 of the Command register is set to 1. Software clears this bit by writing a 1 to this bit location.
7	1h RO	**Fast Back-to-Back Capable (FBBC):** Reserved as 1 Read-Only.
6	0h RO	**User Definable Features (UDF):** Reserved as 0. Read-Only.
5	0h RO	**66 MHz Capable (MC):** Reserved as 0. Read-Only.
4	1h RO	**Capabilities List (CL):** Hardwired to 1 indicating that offset 34h contains a valid capabilities pointer.
3	0h RO/V	**Interrupt Status (IS):** This read-only bit reflects the state of this function's interrupt at the input of the enable/disable logic. This bit is a 1 when the interrupt is asserted. This bit will be 0 when the interrupt is deasserted. The value reported in this bit is independent of the value in the Interrupt Enable bit.
2:0	0h RO	**Reserved (RSVD)**

16.311 Revision ID (RID_MMIO)—Offset 8608h

Access Method

Type: MEM Register
(Size: 8 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO/V	**Revision ID (RID):** See Chap 6 for value.

16.312 Programming Interface (PI_MMIO)—Offset 8609h

Access Method

Type: MEM Register
(Size: 8 bits)

Device:
Function:

Default: 30h

Bit Range	Default & Access	Field Name (ID): Description
7:0	30h RO	**Programming Interface (PI):** A value of 30h indicates that this USB Host Controller conforms to the XHCI specification.

16.313 Sub Class Code (SCC_MMIO)—Offset 860Ah

Access Method

Type: MEM Register
(Size: 8 bits)

Device:
Function:

Default: 3h

Bit Range	Default & Access	Field Name (ID): Description
7:0	3h RO	**Sub Class Code (SCC):** A value of 03h indicates that this is a Universal Serial Bus Host Controller.

16.314 Base Class Code (BCC_MMIO)—Offset 860Bh

Access Method

Type: MEM Register
(Size: 8 bits)

Device:
Function:

Default: Ch

Bit Range	Default & Access	Field Name (ID): Description
7:0	Ch RO	**Base Class Code (BCC):** A value of 0Ch indicates that this is a Serial Bus controller.

16.315 Master Latency Timer (MLT_MMIO)—Offset 860Dh

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Master Latency Timer (MLT):** Because the XHC controller is internally implemented with arbitration on an internal interface, it does not need a master latency timer. The bits will be fixed at 0.

16.316 Header Type (HT_MMIO)—Offset 860Eh

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 80h

Bit Range	Default & Access	Field Name (ID): Description
7	1h RO	**Multi-Function Bit (MFB):** Read only indicating single function device.
6:0	0h RO	**Configuration layout (CL):** Hardwired to 0 to indicate a standard PCI configuration layout.

16.317 Memory Base Address (MBAR_MMIO)—Offset 8610h

Dummy register, mirror of physical register as MBAR. Value in this register will be different after the enumeration process.

Access Method

Type: MEM Register (Size: 64 bits)
Device:
Function:

Default: 4h

Bit Range	Default & Access	Field Name (ID): Description
63:16	0h RW	**Base Address (BA):** Bits (63:16) correspond to memory address signals (63:16), respectively. This gives 64 KB of relocatable memory space aligned to 64 KB boundaries.
15:4	0h RO	**Reserved (RSVD):** Reserved. Read-Only 0, this indicates that this function is requesting an 64KB block of memory.
3	0h RO	**Prefetchable (Prefetchable):** This bit is hardwired to 0 indicating that this range should not be prefetched.
2:1	2h RO	**Type (Type):** If this field is hardwired to 00 it indicates that this range can be mapped anywhere within 32-bit address space. If this field is hardwired to 10 it indicates that this range can be mapped anywhere within 64-bit address space.
0	0h RO	**Resource Type Indicator (RTE):** This bit is hardwired to 0 indicating that the base address field in this register maps to memory space

16.318 USB Subsystem Vendor ID (SSVID_MMIO)—Offset 862Ch

Dummy register, mirror of physical register as SSVID. This register is modified and maintained by BIOS

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW/L	**USB Subsystem Vendor ID (SSVID):** This register, in combination with the USB Subsystem ID register, enables the operating system to distinguish each subsystem from the others.

16.319 USB Subsystem ID (SSID_MMIO)—Offset 862Eh

Dummy register, mirror of physical register as SSID. This register is modified and maintained by BIOS

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW/L	**USB Subsystem ID (SSID):** BIOS sets the value in this register to identify the Subsystem ID. This register, in combination with the Subsystem Vendor ID register, enables the operating system to distinguish each subsystem from other(s).

16.320 Capabilities Pointer (CAP_PTR_MMIO)—Offset 8634h

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 70h

Bit Range	Default & Access	Field Name (ID): Description
7:0	70h RO	**Capabilities Pointer (CAP_PTR):** This register points to the starting offset of the capabilities ranges.

16.321 Interrupt Line (ILINE_MMIO)—Offset 863Ch

Dummy register, mirror of physical register as ILINE.

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Interrupt Line (ILINE):** This data is not used by the Intel PCH. It is used as a scratchpad register to communicate to software the interrupt line that the interrupt pin is connected to.

16.322 Interrupt Pin (IPIN_MMIO)—Offset 863Dh

Dummy register, mirror of physical register as IPIN.

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW/L	**Interrupt pin (IPIN):** Bits 7:0 reflect the Interrupt Pin assigned to the host controller by the platform (and are hardwired).

16.323 XHC System Bus Configuration 1 (XHCC1_MMIO)—Offset 8640h

Dummy register, mirror of physical register as XHCC1

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 1FDh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/O	**Access Control (ACCTRL):** This bit is used by BIOS to lock/unlock lockable bits. When set to '1' the write access to bits locked by this bit is disabled (locked state). When set to '0', the write access to bit locked by this bit is enabled (unlocked state). Writable once after platform reset.
30:25	0h RW	**ECO1 (ECO1):** Reserved
24	0h RW	**Master/Target Abort SERR (RMTASERR):** When set, it allows the out-of-band error reporting from the xHCI Controller to be reported as SERR# (if SERR# reporting is enabled) and thus set the STS.SSE bit.
23	0h RW/C	**Unsupported Request Detected (URD):** Set the HW when xHCI Controller received an unsupported request posted cycle. Cleared by SW when the bit is written with value of '1'.
22	0h RW	**Unsupported Request Report Enable (URRE):** When set this bit allows the URD bit to be reported as SERR# (if SERR# reporting is enabled) and thus set the STS.SSE bit.
21:19	0h RW	**Inactivity Initiated L1 Enable (IIL1E):** If programmed to non-zero, it allows L1 power managed to be enabled after the time-out period specified. 000: Disabled 001: 32 bb_cclk 010: 64 bb_cclk 011: 128 bb_cclk 100: 256 bb_cclk 101: 512 bb_cclk 110: 1024 bb_cclk 111: 131072 bb_cclk
18	0h RW	**XHC Initiated L1 Enable (XHCIL1E):** If set, allow the XHC initiated L1 power mangement to be enabled.
17	0h RW	**D3 Initiated L1 Enable (D3IL1E):** If set, allow PCI device state D3 initiated L1 power managment to be enables.

Bit Range	Default & Access	Field Name (ID): Description
16:12	0h RW	**Periodic Complete Pre Wake Time (PCPWT):** The value programmed in this field determines how far in advance of the start of the next micro-frame the host controller must de-assert the "Periodic Complete" signal . This allows for platform wake time before the next scheduled periodic transaction. The value programmed in this field represents the # of bytes consumed in the current micro-frame which is required to allow for the periodic complete to de-assert. This allows for a programmable time to cause the periodic complete to de-assert prior to the start of the next micro-frame. Register Format: Bits (16:12) represents the # of bytes remaining with a 256B granularity. Periodic Complete will de-assert if the bytes consumed in the current micro-frame is less
11	0h RW	**SW Assisted xHC Idle (SWAXHCI):** This bit being set will indicate xHC idleness (through SW means), which must be a '1' to allow L1 entry, and subsequently allow backbone clock to be gated. This bit is to be set by Intel xHCI driver after checking that the xHCI Controller will stay in idle state for a significant period of time, e.g. all ports disconnected. This bit can be cleared under the following conditions (see SWAXHCI Policy bits in xHC System Bus Configuration 2 register): n SW: SW could write 0 to clear this bit. n HW: HW, under policy control, will clear this bit on an MMIO access to the Host Controller. n HW: HW, under policy control, will clear this bit when HW exits Idle state.
10:8	1h RW	**L23 to Host Reset Acknowledge Wait Count (L23HRAWC):** If programmed to non zero, it allows a wait period after the L23 PHY has shutdown before returning host reset acknowledge to PMC. 000: Disabled 001: 128 bb_cclk 010: 256 bb_cclk 011: 512 bb_cclk 100: 1024 bb_cclk 101: 2048 bb_cclk 110: 4096 bb_cclk 111: 131072 bb_cclk
7:6	3h RW	**Upstream Type Arbiter Grant Count Posted (UTAGCP):** Grant count for IOSF upstream L2 request type arbiter for posted type
5:4	3h RW	**Upstream Type Arbiter Grant Count Non Posted (UDAGCNP):** Grant count for IOSF upstream L2 type arbiter for non-posted type
3:2	3h RW	**Upstream Type Arbiter Grant Count Completion (UDAGCCP) (UDAGCCP):** Grant count for IOSF upstream L2 type arbiter for completion type
1:0	1h RW	**Upstream Device Arbiter Grant Count (UDAGC) (UDAGC):** Grant count for IOSF upstream L1 device arbiter

16.324 XHC System Bus Configuration 2 (XHCC2_MMIO)—Offset 8644h

Dummy register, mirror of physical register as XHCC2

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 3C000h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**OC Configuration Done (OCCFGDONE):** This bit is used by BIOS to prevent spurious switching during OC configuration. It must be set by BIOS after configuration of the OC mapping bits is complete. Once this bit is set, OC mapping shall not be changed by SW.
30:26	0h RW	**ECO1 (ECO1):** Reserved.
25	0h RW	**DMA Request Boundary Crossing Control (DREQBCC):** This bit controls the boundary crossing limit of each Read/Write Request. 0: 4KB 1: 64B
24:22	0h RW	**IDMA Read Request Size Control (IDMA_RDREQSZCTRL):** Read Request Size Control: This bit controls the maximum size of each Read Request. 000: 128B 001: 256B 011 - 110: Reserved 111: 64B
21	0h RW	**XHC Upstream Read Relaxed Ordering Enable (XHCUPRDROE):** This policy controls the Relaxed Ordering attribute for upstream reads. 0 - xHC will clear RO for all upstream read requests. 1 - xHC will set RO for all upstream read requests.
20	0h RW	**IOSF Sideband Register Access Disable (IOSFSRAD):** When set, it disables the IOSF sideband interface from accepting any host space register access.
19:14	Fh RW	**Upstream Non-Posted Pre-Allocation (UNPPA):** This field reserves data sizes, in 64 byte chunks, of the downstream completion resource. This value is zero based. 000000 - 111111: Pre-allocate 64 bytes - 4096 bytes If set greater than the default allows over-allocation If set less than default allows under-allocation Only allowed to be programmed when BME = 0 and no outstanding downstream completion
13:12	0h RW	**SW Assisted xHC Idle Policy (SWAXHCIP):** Note: Irrespective of the setting of this field, SW write of 0 to SWAXHCI will clear the bit. 00b (default): xHC HW clears SWAXHCI bit upon: n MMIO access to Host Controller OR n xHC HW exits Idle state 01b: xHC HW does not autonomously clear SWAXHCI bit. The bit could be cleared only by SW. 10b: xHC HW clears SWAXHCI upon MMIO acces to Host Controller. xHC HW exit from Idle state will not clear SWAXHCI. 11b: Reserved
11	0h RW	**MMIO Read After MMIO Write Delay Disable (RAWDD):** This field controls delay on MMIO Read after MMIO Write. 0b (Default): Delay MMIO Read after MMIO Write 1b: Do not delay MMIO Read after MMIO Write Note that this delay applies after the second of the two DW writes in the case where the IOSF Gasket splits a QW write into two single DW writes to the IP.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW	**MMIO Write After MMIO Write Delay Enable (WAWDE):** This field controls delay on MMIO Write after previous MMIO Write. 0b (Default): Do not delay MMIO Write after previous MMIO Write 1b: Delay MMIO Write after previous MMIO Write Note that the delay count does not apply on the second of the two DW writes that are generated by IOSF Gasket when it splits a QW write into two. In other words, the second of the two DW writes could happen without any delay with respect to the first DW write. This choice is being made for ease of ECO. The delay count, in this case, will apply after the second of the two DW writes.
9:8	0h RW	**SW Assisted Cx Inhibit (SWACXIHB):** This field controls how the DMI L1 inhibit signal from USB3 to PMC will behave. 00: Never inhibit Cx 01: Inhibit Cx when Isochronous Endpoint is active (PPT Behavior) 10: Inhibit Cx when Periodic Active as defined in 40.4.3.2.1 11: Always inhibit Cx
7:6	0h RW	**SW Assisted DMI L1 Inhibit (SWADMIL1IHB):** This field controls how the DMI L1 inhibit signal from USB3 to DMI will behave. 00: Never inhibit DMI L1. 01: Inhibit DMI L1 when Isochronous Endpoint is active (PPT Behavior). 10: Inhibit DMI L1 when Priodic Active as defined in 40.4.3.2.1. 11: Inhibit DMI L1 if XHCC1.SWAXHCI = 0.
5:3	0h RW	**L1 Force P2 Clock Gating Wait Count (L1FP2CGWC):** If programmed to non zero, it allows L1 force P2 gating off the clock to be delayed after the time-out period specified. If wake up event is detected before the time-out, pclk remains alive and trigger L1 exit as though CPU host is causing the wake, 000: Disabled 001: 128 bb_cclk 010: 256 bb_cclk 011: 512 bb_cclk 100: 1024 bb_cclk 101: 2048 bb_cclk 110: 4096 bb_cclk 111: 131072 bb_cclk
2:0	0h RW	**Read Request Size Control (RDREQSZCTRL):** Read Request Size Control: This bit controls the maximum size of each Read Request. 000: 128B 001: 256B 010: 512B 011 - 110: Reserved 111: 64B

16.325 Clock Gating (XHCLKGTEN_MMIO)—Offset 8650h

Dummy register, mirror of physical register as XHCLKGTEN

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**Rsvd2 (Rsvd2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
28	0h RW	**Nak'ing USB2.0 EPs for Backbone Clock Gating and PLL Shutdown (NUEFBCGPS):** This field controls whether Naking USB2.0 EPs, once in Naking low priority schedule, should be considered as active for the considerations for backbone clock gating and PLL shutdown or not. 0: Naking USB2.0 EPs are not considered to be active for Backbone Clock Gating and PLL Shutdown evaluation. 1: Naking USB2.0 EPs are considered to be active for Backbone clock gating and PLL shutdown evaluation.
27	0h RW	**SRAM Power Gate Enable (SRAMPGTEN):** This register enables the SRAM Power Gating when PLL shutdown conditions for all clock domains have been met 0 - Disallow SRAM Power Gating. 1 - Allow SRAM Power Gating
26	0h RW	**SS Link PLL Shutdown Enable (SSLSE):** This register enables the SS P3 state to be exposed to PXP PLL Shutdown conditions on behalf of all USB SS Ports ontop of trunk clock gating. 0 - P3 state NOT allowed to result in PXP PLL shutdown. 1- P3 state allowed to result in PXP PLL shutdown
25	0h RW	**USB2 PLL Shutdown Enable (USB2PLLSE):** When set, this bit allows USB2 PLL to be shutdown when HS Link trunk clock is gated, and xHC can tolerate PLL spin up time for subsequent clock request. Note: if USB2 PLL Shutdown Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value.
24	0h RW	**IOSF Sideband Trunk Clock Gating Enable (IOSFSTCGE):** When set, this bit allows the IOSF sideband clock trunk to be gated when idle conditions are met.
23:20	0h RW	**HS Backbone PXP Trunk Clock Gate Enable (HSTCGE):** This register determines the HS Ux state(s) which will be exposed to Backbone PXP trunk gating of core clock. Uy is a state allowed to result in trunk gating when ss_tcg_ux_en(x) is asserted. (0) ==) U0 or deeper (1) ==) NA (no support for U1) (2) ==) U2 (L1) or deeper (3) ==) U3 (L2) or deeper
19:16	0h RW	**SS Backbone PXP Trunk Clock Gate Enable (SSTCGE):** This register determines the SS Ux state(s) which will be exposed to Backbone PXP trunk gating of core clock. Uy is a state allowed to result in trunk gating when ss_tcg_ux_en(x) is asserted. (0) ==) U0 or deeper (1) ==) U1 or deeper (2) ==) U2 or deeper (3) ==) U3 or deeper
15	0h RW	**XHC Ignore_EU3S (XHCIGEU3S):** This register determines if the xHC will use the EU3S as a condition to allow for Frame timer gating. 0 - xHC may allow frame timer to be gated when EU3S is set and all ports are in the required state. 1 - xHC may allow frame timer to be gated regardless of EU3S.
14	0h RW	**XHC Frame Timer Clock Shutdown Enable (XHCFTCLKSE):** This register determines if the xHC will allow the frame timer clock to be gated. 0 - xHC will not allow ICC PLL 96MHz output to be shutdown thus keeping the frame timer running. 1 - xHC will allow ICC PLL 96MHz output to be shutdown under specific conditions.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**XHC Backbone PXP Trunk Clock Gate In Presence of ISOCH EP (XHCBBTCGIPISO):** This register controls the policy on allowing Backbone PXP trunk clock gate in the presence of IDLE ISOCH EP s with active DB. Allows the periodic active to be used in enabling Backbone PXP trunk clock gating of core clock. 0 Trunk gate of core clock is not allowed when ISOCH EP DB is set and ISOCH EP s are idle. 1 Allow trunk gate of core clock when ISOCH EP DB is set and ISOCH EP s are idle.
12	0h RW	**XHC HS Backbone PXP Trunk Clock Gate U2 non RWE (XHCHSTCGU2NRWE):** This register controls the policy on allowing Backbone PXP trunk gating of core clock when there is atleast 1 non Remote Wake Enabled HS Port in U2. 0 Prevent trunk gate of core clock when a non RWE HS Port is in U2. 1 Allow trunk gate of core clock when a non RWE HS Port is in U2.
11:10	0h RW	**XHC USB2 PLL Shutdown Lx Enable (XHCUSB2PLLSDLE):** This register determines the HS Link state(s) which will be exposed to USB2 PLL Shutdown conditions on behalf of all USB2 HS Ports. Ly is a state allowed to result in USB2 PLL Shutdown when en(x) is asserted. (0) ==) L1 or deeper (1) ==) L2 or deeper
9:8	0h RW	**HS Backbone PXP PLL Shutdown Ux Enable (HSUXDMIPLLSE):** This register determines the Ux state(s) which will be exposed to PXP PLL Shutdown conditions. PLL Shutdown is allowed in: 00b Disabled (Link states shall be disabled for DMI PLL shutdown) 01b U0 or conditions for 10b setting. 10b U2 or conditions for 11b setting. 10b U3, Disconnected, Disabled or Powered-Off.
7:5	0h RW	**SS Backbone PXP PLL Shutdown Ux Enable (SSPLLSUE):** This register determines the Ux state(s) which will be exposed to DMI PLL Shutdown conditions. PLL Shutdown is allowed in: 000b Disabled (Link states shall be ignored for DMI PLL shutdown). 001b U0 or conditions for 010b setting 010b U1 or conditions for 011b setting 011b U2 or conditions for 100b setting 100b U3, Disconnected, Disabled or Powered-Off
4	0h RW	**XHC Backbone Local Clock Gating Enable (XHCBLCGE):** When set, this bit allows XHCI Controller IP backbone clock to be locally gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value.
3	0h RW	**HS Link Trunk Clock Gating Enable (HSLTCGE):** When set, this bit allows High Speed Link control's 480 MHz and its 48/60 MHz link clock trunk to be gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value. BIOS need to query fuses to see if HW is enabled.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**SS Link Trunk Clock Gating Enable (SSLTCGE):** When set, this bit allows the SuperSpeed Link control's 250 MHz and its divided 125 MHz link clock trunk to ge gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value. BIOS need to query fuses to see if HW is enabled.
1	0h RW	**IOSF Backbone Trunk Clock Gating Enable (IOSFBTCGE):** When set, this bit allows the IOSF backbone clock trunk to be gated when idle conditions are met.
0	0h RW	**IOSF Gasket Backbone Local Clock Gating Enable (IOSFBLCGE):** When set, this bit allows the IOSF Gasket backbone clock to be locally gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value. BIOS need to query fuses to see if HW is enabled.

16.326 Audio Time Synchronization (AUDSYNC_MMIO)—Offset 8658h

This 32 bit register is used for audio stream synchronization across different devices. Global signal sample_now captures a value in AUDSYNC register.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Rsvd2 (Rsvd2)**
29:16	0h RO/V	**Captured Frame List Current Index/Frame Number (CMFI):** The value in this register is updated in response to sample_now signal. Bits (29:16) reflect state of bits (13:0) of FRINDEX
15:13	0h RO	**Rsvd1 (Rsvd1)**
12:0	0h RO/V	**Captured Micro-frame BLIF (CMFB):** The value is updated in response to sample_now signal and provides information about offset within micro-frame. Captured value represents number of 8 high-speed bit time units from start of micro-frame. At the beginning of micro-frame captured value will be 0 and increase to maximum value at the end. Default maximum value is 7499 but it may be changed as result of adjustment done in FLA.

16.327 Serial Bus Release Number (SBRN_MMIO)—Offset 8660h

Access Method

Type: MEM Register
(Size: 8 bits)

Device:
Function:

Default: 30h

Bit Range	Default & Access	Field Name (ID): Description
7:0	30h RO	**Serial Bus Release Number (SBRN):** A value of 30h indicates that this controller follows USB release 3.0.

16.328 Frame Length Adjustment (FLADJ_MMIO)—Offset 8661h

This feature is used to adjust any offset from the clock source that generates the clock that drives the SOF counter. When a new value is written into these six bits, the length of the frame is adjusted. Its initial programmed value is system dependent based on the accuracy of hardware USB clock and is initialized by system BIOS. This register should only be modified when the HChalted bit in the USBSTS register is a one. Changing value of this register while the host controller is operating yields undefined results.

Access Method

Type: MEM Register
(Size: 8 bits)

Device:
Function:

Default: 60h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**Reserved (RSVD):** Read-Only. These bits are reserved for future use and should read as "00".
6	1h RO	**No Frame Length Timing Capability (NO_FRAME_LENGTH_TIMING_CAP):** This flag is set to 1 to indicate that the host controller does not support a programmable Frame Length Timing Value field.
5:0	20h RO	**Frame Length Timing Value (FLTV):** SOF micro-frame length) is equal to 59488 + value in this field. The default value is decimal 32 (20h), which gives a SOF cycle time of 60000. Frame Length (number of High Speed bit times) FLADJ Value (decimal) (decimal) 59488 0 (00h) 59504 1 (01h) 59520 2 (02h) ... 59984 31 (1Fh) 60000 32 (20h) ... 60480 62 (3Eh) 60496 63 (3Fh) Each decimal value change to this register corresponds to 16 high-speed bit times. The SOF cycle time (number of SOF counter clock periods to generate a SOF micro-frame length) is equal to 59488 + value in this field. The default value is decimal 32 (20h), which gives a SOF cycle time of 60000. Frame Length (# High Speed bit times) FLADJ Value

16.329 Best Effort Service Latency (BESL_MMIO)—Offset 8662h

Dummy register, mirror of physical register as BESL. Bset Effort Service Latency.

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:4	0h RW/L	**Default Best Effort Service Latency Deep (DBESLD):** Default Best Effort Service Latency (DBESLD) If the value of this field is non-zero, it defines the recommended value for programming the PORTPMSC register BESLD field. This is programmed by BIOS based on platform parameters.
3:0	0h RW/L	**Default Best Effort Service Latency (DBESL):** If the value of this field is non-zero, it defines the recommended value for programming the PORTPMSC register BESL field. This is programmed by BIOS based on platform parameters.

16.330 PCI Power Management Capability ID (PM_CID_MMIO)—Offset 8670h

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
7:0	1h RO	**PCI Power Management Capability ID (PM_CID):** A value of 01h indicates that this is a PCI Power Management capabilities field.

16.331 Next Item Pointer #1 (PM_NEXT_MMIO)—Offset 8671h

Dummy register, mirror of physical register as PM_NEXT. This register is modified and maintained by BIOS

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 80h

Bit Range	Default & Access	Field Name (ID): Description
7:0	80h RW/L	**Next Item Pointer #1 (PM_NEXT):** This register defaults to 80h, which indicates that the next capability registers begin at configuration offset 80h. This register is writable when the Access Control bit is set to '0'. This allows BIOS to effectively hide the next capability registers, if necessary. This register should only be written during system initialization before the plug-and-play software has enabled any master-initiated traffic. Values of: 80h implies next capability is MSI 00h implies that MSI capability is hidden. Note: This value is never expected to be programmed.

16.332 Power Management Capabilities (PM_CAP_MMIO)—Offset 8672h

Dummy register, mirror of physical register as PM_CAP. Normally, this register is read-only to report capabilities to the power management software. In order to report different power management capabilities depending on the system in which the Intel PCH is used, the write access to this register is controlled by the Access Control bit (ACCTRL). The value written to this register does not affect the hardware other than changing the value returned during a read. This register is modified and maintained by BIOS

Access Method

Type: MEM Register (Size: 16 bits)

Device:
Function:

Default: C1C2h

Bit Range	Default & Access	Field Name (ID): Description
15:11	18h RW/L	**PME_Support (PME_Support):** This 5-bit field indicates the power states in which the function may assert PME#. The Intel PCH XHC does not support the D1 or D2 states. For all other states, the Intel PCH XHC is capable of generating PME#. Software should never need to modify this field.
10	0h RW/L	**D2_Support (D2_Support):** The D2 state is not supported.
9	0h RW/L	**D1_Support (D1_Support):** The D1 state is not supported.
8:6	7h RW/L	**Aux_Current (Aux_Current):** The Intel PCH XHC reports 375mA maximum Suspend well current required when in the D3cold state. This value can be written by BIOS when a more accurate value is known.
5	0h RW/L	**DSI (DSI):** The Intel PCH reports 0, indicating that no device-specific initialization is required.
4	0h RO	**Reserved (RSVD)**

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW/L	**PME Clock (PMEClock):** The Intel PCH reports 0, indicating that no PCI clock is required to generate PME#.
2:0	2h RW/L	**Version (Version):** The Intel PCH reports 010, indicating that it complies with Revision 1.1 of the PCI Power Management Specification.

16.333 Power Management Control/Status (PM_CS_MMIO)—Offset 8674h

Dummy register, mirror of physical register as PM_CS

Access Method

Type: MEM Register (Size: 16 bits)

Device:

Function:

Default: 8h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**PME_Status (PME_Status):** This bit is set when the Intel PCH XHC would normally assert the PME# signal independent of the state of the PME_En bit. Writing a 1 to this bit will clear it and cause the internal PME to deassert (if enabled). Writing a 0 has no effect. This bit must be explicitly cleared by the operating system each time the operating system is loaded.
14:13	0h RO	**Data_Scale (Data_Scale):** The Intel PCH hardwires these bits to 00 because it does not support the associated Data register.
12:9	0h RO	**Data_Select (Data_Select):** The Intel PCH hardwires these bits to 0000 because it does not support the associated Data register.
8	0h RW	**PME_En (PME_En):** A 1 enables the Intel PCH XHC to generate an internal PME signal when PME_Status is 1. This bit must be explicitly cleared by the operating system each time it is initially loaded.
7:4	0h RO	**Reserved (RSVD)**
3	1h RO	**No Soft Reset (NSR):** No_Soft_Reset - When set ("1"), this bit indicates that devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset. Configuration Context is preserved. Upon transition from the D3hot to the D0 Initialized state, no additional operating system intervention is required to preserve Configuration Context beyond writing the PowerState bits. Transition from D3hot to D0 by a system or bus segment reset will return to the device state D0 Uninitialized with only PME context preserved if PME is supported and enabled.
2	0h RO	**Reserved (RSVD2)**

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**PowerState (PowerState):** This 2-bit field is used both to determine the current power state of XHC function and to set a new power state. The definition of the field values are: 00b - D0 state 11b - D3hot state If software attempts to write a value of 10b or 01b in to this field, the write operation must complete normally, however, the data is discarded and no state change occurs. When in the D3hot state, the Intel PCH must not accept accesses to the XHC memory range, but the configuration space must still be accessible.

16.334 Message Signaled Interrupt CID (MSI_CID_MMIO)—Offset 8680h

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 5h

Bit Range	Default & Access	Field Name (ID): Description
7:0	5h RO	**Capability ID (CID):** Indicates that this is an MSI capability

16.335 Next item pointer (MSI_NEXT_MMIO)—Offset 8681h

Dummy register, mirror of physical register as MSI_NEXT

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW/L	**Next Pointer (NEXT):** Indicates that this is the last item on the capability list

16.336 Message Signaled Interrupt Message Control (MSI_MCTL_MMIO)—Offset 8682h

Dummy register, mirror of physical register as MSI_MCTL

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 86h

Bit Range	Default & Access	Field Name (ID): Description
15:9	0h RO	**Reserved (RSVD)**
8	0h RO	**Per-Vector Masking Capable (PVM):** Specifies whether controller supports MSI per vector masking. Not supported
7	1h RO	**64 Bit Address Capable (C64):** Specifies whether capable of generating 64-bit messages. This device is 64-bit capable.
6:4	0h RW	**Multiple Message Enable (MME):** Indicates the number of messages the controller should assert. This device supports multiple message MSI.
3:1	3h RO	**Multiple Message Capable (MMC):** Indicates the number of messages the controller wishes to assert. This field must be set by HW to reflect the number of Interrupters supported. Encoding number of Vectors requested (number of Interrupters) 000 1 001 2 010 4 011 8 100 16 101 32 110-111 Reserved
0	0h RW	**MSI Enable (MSIE):** If set to 1, MSI is enabled and the traditional interrupt pins are not used to generate interrupts. If cleared to 0, MSI operation is disabled and the traditional interrupt pins are used.

16.337 Message Signaled Interrupt Message Address (MSI_MAD_MMIO)—Offset 8684h

Dummy register, mirror of physical register as MSI_MAD

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**Addr (Addr):** Lower DW of system specified message address, always DWORD aligned
1:0	0h RO	**Reserved (RSVD)**

16.338 Message Signaled Interrupt Upper Address (MSI_MUAD_MMIO)—Offset 8688h

Dummy register, mirror of physical register as MSI_MUAD

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Upper Addr (UpperAddr):** Upper DW of system specified message address.

16.339 Message Signaled Interrupt Message Data (MSI_MD_MMIO)—Offset 868Ch

Dummy register, mirror of physical register as MSI_MD

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW	**Data (Data):** This 16-bit field is programmed by system software if MSI is enabled. Its content is driven onto the lower word (PCI AD(15:0)) during the data phase of the MSI memory write transaction. The Multiple Message Enable field (bits 6-4 of the Message Control register) defines the number of low order message data bits the function is permitted to modify to generate its system software allocated vectors. For example, a Multiple Message Enable encoding of 010 indicates the function has been allocated four vectors and is permitted to modify message data bits 1 and 0 (a function modifies the lower message data bits to generate the allocated number of vectors). If the Multiple Message Enable field is 000, the function is not permitted to modify the message data.

16.340 Device Idle Capability (DEVIDLE_MMIO)—Offset 8690h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: F0140009h

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RO	**VID (VID)**

Bit Range	Default & Access	Field Name (ID): Description
27:24	0h RO	**REV (REV)**
23:16	14h RO	**Length (LENGTH):** Indicates that this capability is 16 bytes long.
15:8	0h RO	**Next Capability Pointer (NCP):** This field contains the offset to the next PCI Capability structure or 000h if no other items exist in the linked list of Capabilities.
7:0	9h RO	**Capability ID (CID)**

16.341 Vendor Specific Header (VSHDR_MMIO)—Offset 8694h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 1400010h

Bit Range	Default & Access	Field Name (ID): Description
31:20	14h RO	**VSEC Length (VSEC_LENGTH):** This field indicates the number of bytes in the entire VSEC structure, including the PCI Extended Capability header, the Vendor- Specific header, and the Vendor-Specific register
19:16	0h RO	**VSEC Rev (VSEC_REV):** This field is a vendor-defined version number that indicates the version of the VSEC structure. Software must qualify the Vendor ID and VSEC ID before interpreting this field.
15:0	10h RO	**VSEC ID (VSEC_ID):** This field is a vendor-defined ID number that indicates the nature and format of the VSEC structure. Software must qualify the Vendor ID before interpreting this field.

16.342 SW LTR POINTER (SWLTRPTR_MMIO)—Offset 8698h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**SW LTR Update MMIO Offset Location (SW_LTR_UPDT_MMIO_OFFSET):** This register contains the location pointer to the SW LTR register in MMIO space, as an offset from the specified BAR. The value of this register is a don't care, if the Valid bit is not set.
3:1	0h RO	**Base Address Register Number (BARNUM):** Contains the 0's based AR Number of the BAR which contains the location of the SW LTR MMIO register. When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number. Bar 0 is the 1st BAR. The value of this register is a don't care, if the Valid bit is not set.
0	0h RO	**Valid (VALID):** Set to '1' to indicate that the function has implemented a SW LTR register as specified in the DevIdle that can be located using the SWLTRLOC register and BARNUM. Set to '0' to indicate that the function has not implemented a SW LTR register that is compliant to the DevIdle definition. This could be because the function has not implemented SW LTR at all, or has a device specific version of SW LTR.

16.343 Device Idle Pointer Register (DEVIDLEPTR_MMIO)—Offset 869Ch

Dummy register, mirror of physical register as DEVIDLEPTR

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 80AC1h

Bit Range	Default & Access	Field Name (ID): Description
31:4	80ACh RW/L	**DevIdle MMIO Offset Location (DEVIDLELOC):** This register contains the location pointer to the DevIdle register in MMIO space, as an offset from the specified BAR. The value of this register is a don't care, if the Valid bit is not set.
3:1	0h RO	**Base Address Register Number (BARNUM):** Contains the 0's based BAR Number of the BAR which contains the location of the DevIdle MMIO register. When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number. Bar 0 is the 1st BAR. The value of this register is a don't care, if the Valid bit is not set.

Bit Range	Default & Access	Field Name (ID): Description
0	1h RW/L	**Valid (VALID):** Set to '1' to indicate that the function has implemented a DevIdle register as specified in the DevIdle that can be located using the DEVIDLELOC register and BARNUM. Set to '0' to indicate that the function has not implemented a DevIdle register that is compliant to the DevIdle definition. This could be because the function has not implemented DevIdle at all, or has a device specific version of DevIdle.

16.344 Device Idle Power ON Latency (DEVIDLEPOL_MMIO)—Offset 86A0h

Dummy register, mirror of physical register as DEVIDLEPOL

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 800h

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	**Rsvd1 (Rsvd1)**
12:10	2h RW/L	**Power On Latency Scale (POLS):** Latency Scale multiplier: 010: 1us 011: 32us All other settings are reserved. The value of this register is multiplied with the Power On Latency Value (POLV) to provide the DevIdle power on exit latency multiplier scale from 1us to 32ms. This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value. This register is undefined if the DevIdle.Valid bit is '0'.
9:0	0h RW/L	**Power On Latency Value (POLV):** 10-bit value that is multiplied by the Power On Latency Scale. A value of 0 indicates a power on latency of less than 1us. This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value. This register is undefined if the DevIdle.Valid bit is '0'.

16.345 High Speed Configuration 2 (HSCFG2_MMIO)—Offset 86A4h

Dummy register, mirror of physical register as HSCFG2

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2000h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Rsvd1 (Rsvd1)**
18	0h RW	**PORT1_HOST_MODE_OVERRIDE (PORT1_HOST_MODE_OVERRIDE):** When set, this bit causes the Host_Device mux on port 1 to be forced into the Host mode.
17:16	0h RW	**eUSB2SEL (eUSB2SEL):** The two bits are associate with USB2 ports 1 - bit 16 and 2 - bit 2 0: Port is mapped to USB2 1: Port is mapped to eUSB2
15	0h RW	**HS ASYNC Active IN Mask (HSAAIM):** Determines if the Async Active will mask/ignore IN EP s. 0 HS ASYNC Active will include IN EP s. 1 HS ASYNC Active will mask/ignore IN EP s.
14	0h RW	**HS OUT ASYNC Active Polling EP Mask (HSOAAPEPM):** Determines if the Async Active for OUT HS/FS/LS masks/ignores EP s that are polling/PINGing (HS) due to NAK. 0 HS OUT ASYNC Active will include EP s that are polling. 1 HS OUT ASYNC Active will mask/ignore EP s that are polling.
13	1h RW	**HS IN ASYNC Active Polling EP Mask (HSIAAPEPM):** Determines if the Async Active for IN HS/FS/LS masks/ignores EP s that are polling due to NAK. 0 HS IN ASYNC Active will include EP s that are polling. 1 HS IN ASYNC Active will mask/ignore EP s that are polling.
12:11	0h RW	**HS INTR IN Periodic Active Policy Control (HSIIPAPC):** Controls how the HS INTR IN periodic active is used to generate the global periodic active. This will determine how the smallest service interval among active EP s and number of active EP s are used. 0 HS INTR IN periodic active will be used to generate periodic active if Service Interval Threshold OR Numb of EP Threshold values meet the requirement. 1 HS INTR IN periodic active will be used to generate periodic active if Service Interval Threshold AND Numb of EP Threshold values meet the requirement. 2 Always allow HS INTR EP s to be used in the generation of the global Periodic Active indication. 3 Never allow HS INTR EP s to be used in the generation of the global Periodic Active indicaiton
10:4	0h RW	**HS INTR IN Periodic Active Num of EP Threshold (HSIIPANEPT):** Defines the threshold used to determine if Periodic Acive may include HS/FS/LS INTR IN EP active indication. If there are more than NumEPThreshold active HS/FS/LS INTR EP s then they may be included as part of the periodic active generation.
3:0	0h RW	**HS INTR IN Periodic Active Service Interval Threshold (HSIIPASIT):** Defines the Service Interval threshold used to determine if Periodic Acive will include HS/FS/LS INTR IN EP active indication. If there are any active HS/FS/LS INTR EP s with a service interval less than or equal to this threshold then they may be included as part of the periodic active generation.

16.346 XHCI USB2 Overcurrent Pin Mapping 1 (U2OCM1_MMIO)—Offset 86B0h

The RW/L property of this register is controlled by OCCFDONE bit.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Rsvd (Rsvd):** Reserved
8:0	0h RW/L	**OC Mapping (OCM)**

16.347 XHCI USB2 Overcurrent Pin Mapping 2 (U2OCM2_MMIO)—Offset 86B4h

Reserved

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Rsvd (Rsvd):** Reserved
8:0	0h RW/L	**OC Mapping (OCM)**

16.348 XHCI USB3 Overcurrent Pin Mapping 1 (U3OCM1_MMIO)—Offset 86D0h

The RW/L property of this register is controlled by OCCFDONE bit.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Rsvd (Rsvd)**
6:0	0h RW/L	**OC Mapping (OCM)**

16.349 XHCI USB3 Overcurrent Pin Mapping 2 (U3OCM2_MMIO)—Offset 86D4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Rsvd (Rsvd)**
6:0	0h RW/L	**OC Mapping (OCM)**

16.350 XHCC3 (XHCC3_MMIO)—Offset 86FCh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RO	**Reserved (Rsvd1):** Reserved
4	0h RW	**Error Handing: Disable masking of extra read to completion collect buffer (SUPP_XTRA_RD_CPL_COLL_BUF):** When set to 1, the fix to supress extra read to completion collect buffer is disabled. The fix is relevent only when ECC checking is enabled
3	0h RW	**Error Handling : Disable Command Parity Check (DISABLE_COMMAND_PARITY_CHECK):** When set to 1, XHCI Host Controller disables checking of Command Parity on command received as a target on its IOSF Primary interface
2	0h RW	**Error Handling : Disable Data Parity Check (DISABLE_DATA_PARITY_CHECK):** When set to 1, XHCI Host Controller disables checking of Data Parity on data received as a target on its IOSF Primary interface

Bit Range	Default & Access	Field Name (ID): Description
1	1h RW	**Error Handling : Enable ECC Error Response (ENABLE_ECC_ERROR_RESPONSE):** When set to 1, XHCI Host Controller will check for ECC on RFs (that support ECC) and halt operation when uncorrectable ECC is detected
0	0h RW/L	**Function Disable (FXN_DISABLE):** When set will disable the xHC from being operational.

16.351 Debug Capability ID Register (DCID)—Offset 8700h

This register is modified and maintained by BIOS

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 5100Ah

Bit Range	Default & Access	Field Name (ID): Description
31:21	0h RO	**Reserved (RSVD)**
20:16	5h RW	**Debug Capability Event Ring Segment Table Max (DCERSTM):** Note: This register is sticky.
15:8	10h RW	**Next Capability Pointer (NCP):** Note: This register is sticky.
7:0	Ah RW	**Capability ID (CID):** Note: This register is sticky.

16.352 Debug Capability Doorbell Register (DCDB)—Offset 8704h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Reserved (RSVD)**
15:8	0h RW	**Doorbell Target (DBTGT):** This field defines the target of the doorbell reference. The table below defines the Debug Capability notification that is generated by ringing the doorbell. Value Definition 0 Data EP 1 OUT Enqueue Pointer Update 1 Data EP 1 IN Enqueue Pointer Update 2:255 Reserved This field returns '0' when read and the value should be treated as undefined by software.

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Reserved (RSVD_1)**

16.353 Debug Capability Event Ring Segment Table Size Register (DCERSTSZ)—Offset 8708h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Reserved (RSVD)**
15:0	0h RW	**Event Ring Segment Table Size (ERSTS):** This field identifies the number of valid Event Ring Segment Table entries in the Event Ring Segment Table pointed to by the Debug Capability Event Ring Segment Table Base Address register. The maximum value supported by an xHC implementation for this register is defined by the DCERST Max field in the DCID register Software shall initialize this register before setting the Debug Capability Enable field in the DCCTRL register to '1'.

16.354 Debug Capability Event Ring Segment Table Base Address Register (DCERSTBA)—Offset 8710h

Access Method

Type: MEM Register
(Size: 64 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:4	0h RW	**Event Ring Segment Table Base Address Register (ERSTBAR):** This field defines the high order bits of the start address of the Debug Capability Event Ring Segment Table. Software shall initialize this register before setting the Debug Capability Enable field in the DCCTRL register to '1'.
3:0	0h RW	**Reserved (RSVD)**

16.355 Debug Capability Event Ring Dequeue Pointer Register (DCERDP)—Offset 8718h

Access Method

Type: MEM Register
(Size: 64 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:4	0h RW	**Dequeue Pointer (DQP):** This field defines the high order bits of the 64-bit address of the current Debug Capability Event Ring Dequeue Pointer. Software shall initialize this register before setting the Debug Capability Enable field in the DCCTRL register to '1'.
3	0h RW	**Reserved (RSVD)**
2:0	0h RW	**Dequeue ERST Segment Index (DESI):** This field may be used by the xHC to accelerate checking the Event Ring full condition. This field is written with the low order 3 bits of the offset of the ERST entry which defines the Event Ring segment that the Event Ring Dequeue Pointer resides in.

16.356 Debug Capability Control Register (DCCTRL)—Offset 8720h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Debug Capability Enable (DCE)**
30:24	0h RO	**Device Address (DADDR)**
23:16	0h RO	**Debug Max Burst Size (DMBS):** LPT-LP USB Debug Device does not support bursting.
15:5	0h RO	**Reserved (RSVD)**
4	0h RW/C	**DbC Run Change (DRC)**
3	0h RW/1S	**Halt IN TR (HIT)**
2	0h RW/1S	**Halt OUT TR (HOT)**

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Link Status Event Enable (LSE)**
0	0h RO	**DbC Run (DCR)**

16.357 Debug Capability Status Register (DCST)—Offset 8724h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Debug Port Number (DPNUM):** This field provides the ID of the Root Hub port that the Debug Capability has been automatically attached to. The value is '0' when the Debug Capability is not attached to a Root Hub port.
23:1	0h RO	**Reserved (RSVD)**
0	0h RO	**Event Ring Not Empty (ERNE):** When '1', this field indicates that the Debug Capability Event Ring has a Transfer Event on it. It is automatically cleared to '0' by the xHC when the Debug Capability Event Ring is empty, i.e. the Debug Capability Enqueue Pointer is equal to the Debug Capability Event Ring Dequeue Pointer register.

16.358 Debug Capability Port Status and Control Register (DCPORTSC)—Offset 8728h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 80h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD)**
23	0h RW/C	**Port Config Error Change (CEC):** This flag indicates that the port failed to configure its link partner. 0 = No change. 1 = Port Config Error detected. Software shall clear this bit by writing a '1' to it.

Bit Range	Default & Access	Field Name (ID): Description
22	0h RW/C	**Port Link Status Change (PLC):** This flag is set to '1' due to the following PLS transitions: U0 -) U3 Suspend signaling detected from Debug Host U3 -) U0 Resume complete Polling -) Disabled Training Error Ux or Recovery -) Inactive Error Software shall clear this bit by writing a '1' to it. This field is '0' if DCE is '0'
21	0h RW/C	**Port Reset Change (PRC):** This bit is set when reset processing on this port is complete (i.e. a '1' to '0' transition of PR). '0' = No change. '1' = Reset complete.Software shall clear this bit by writing a '1' to it. This field is '0' if DCE is '0'.
20:18	0h RO	**Reserved (RSVD_1)**
17	0h RW/C	**Connect Status Change (CSC):** '1' = Change in Current Connect Status. '0' = No change. Indicates a change has occurred in the port's Current Connect Status. The xHC sets this bit to '1' for all changes to the Debug Device connect status, even if system software has not cleared an existing DbC Connect Status Change. For example, the insertion status changes twice before system software has cleared the changed condition, hardware will be "setting" an already-set bit (i.e., the bit will remain '1'). Software shall clear this bit by writing a '1' to it. This field is '0' if DCE is '0'.
16:14	0h RO	**Reserved (RSVD_2)**
13:10	0h RO	**Port Speed (PSPD):** This field identifies the speed of the port. This field is only relevant when a Debug Host is attached (CCS = '1') in all other cases this field shall indicate Undefined Speed. 0 Undefined Speed 1-15 Protocol Speed ID (PSI) Note: The Debug Capability onlydoes not supports LS, FS, or HS operation.
9	0h RO	**Reserved (RSVD_3)**
8:5	4h RO	**Port Link State (PLS):** This field reflects its current link state. This field is only relevant when a Debug Host is attached (Debug Port Number) '0'). Value Meaning 0 Link is in the U0 State 1 Link is in the U1 State 2 Link is in the U2 State 3 Link is in the U3 State (Device Suspended) 4 Link is in the Disabled State 5 Link is in the RxDetect State 6 Link is in the Inactive State 7 Link is in the Polling State 8 Link is in the Recovery State 9 Link is in the Hot Reset State 15:10 Reserved Note: Transitions between different states are not reflected until the transition is complete.
4	0h RO	**Port Reset (PR):** '1' = Port is in Reset. '0' = Port is not in Reset. This bit is set to '1' when the bus reset sequence as defined in the USB Specification is detected on the Root Hub port assigned to the Debug capability. It is cleared when the bus reset sequence is completed by the Debug Host, and the DbC shall transition to the USB Default state. A '0' to '1' transition of this bit shall clear DCPORTSC PED ('0'). This field is '0' if DCE or CCS are '0'.
3:2	0h RO	**Reserved (RSVD_4)**

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Port Enabled/Disabled (PED):** Default = '0'. '1' = Enabled. '0' = Disabled. This flag shall be set to '1' by a '0' to '1' transition of CCS or a '1' to '0' transition of the PR. When PED transitions from '1' to '0' due to the assertion of PR, the port's link shall transition to the Rx.Detect state. This flag may be used by software to enable or disable the operation of the Root Hub port assigned to the Debug Capability. The Debug Capability Root Hub port operation may be disabled by a fault condition (disconnect event or other fault condition, e.g. a LTSSM Polling substate timeout, tPortConfiguration timeout error, etc.), the assertion of DCPORTSC PR, or by software. 0 = Debug Capability Root Hub port is disabled. 1 = Debug Capability Root Hub port is enabled. When the port is disabled (PED = '0') the port's link shall enter the SS.Disabled state and remain there until PED is reasserted ('1') or DCE is negated ('0'). Note that the Root Hub port is remains mapped to Debug Capability while PED = '0'. While PED = '0' the Debug Capability will appear to be disconnected to the Debug Host. Note, this bit is not affected by PORTSC PR bit transitions. This field is '0' if DCE or CCS are '0'.
0	0h RO	**Current Connect Status (CCS):** '1' = A Root Hub port is connected to a Debug Host and assigned to the Debug Capability. '0' = No Debug Host is present. This value reflects the current state of the port, and may not correspond to the value reported by the Connect Status Change (CSC) field in the Port Status Change Event that was generated by a '0' to '1' transition of this bit. This flag is '0' if Debug Capability Enable (DCE) is '0'.

16.359 Debug Capability Context Pointer Register (DCCP)—Offset 8730h

Access Method

Type: MEM Register (Size: 64 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:4	0h RW	**Debug Capability Context Pointer Register (DCCPR):** This field defines the high order bits of the start address of the Debug Capability Context data structure associated with the Debug Capability. Software shall initialize this register before setting the Debug Capability Enable bit in the Debug Capability Control Register to '1'.
3:0	0h RO	**Reserved (RSVD)**

16.360 Debug Capability Device Descriptor Info Register 1 (DCDDI1)—Offset 8738h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 80870000h

Bit Range	Default & Access	Field Name (ID): Description
31:16	8087h RW	**Vendor ID (VID):** This field is presented by the Debug Device in the USB Device Descriptor idVendor field.
15:8	0h RO	**Reserved (RSVD)**
7:0	0h RW	**DbC Protocol (DBCPR):** This field is presented by the Debug Device in the USB Interface Descriptor bInterfaceProtocol field. Value Function 0 Debug Target vendor defined. 1 GNU Remote Debug Command Set supported. 2-255 Reserved.

16.361 Debug Capability Device Descriptor Info Register 2 (DCDDI2)—Offset 873Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Device Revision (DREV):** This field is presented by the Debug Device in the USB Device Descriptor bcdDevice field.
15:0	0h RW	**Product ID (PID):** This field is presented by the Debug Device in the USB Device Descriptor idProduct field.

16.362 Debug Capability Descriptor Parameters (DCDP)—Offset 8740h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 30C3h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD)**
23:16	0h RW	**Max Power Field (MPF):** This field will be used by USB Debug Device to report maximum power consumption when the device is fully operational. This value is returned by bMaxPower field in response to Configuration Descriptor read from the debug device. Note: bU1DevExitLat and bU2DevExitLat fields returned in BOS Descriptor read will be taken from the corresponding fields from the Host Controller space.
15:8	30h RW/S	**NEXT CAPABILITY POINTER (NCP)**
7:0	C3h RW/S	**Capability ID (CID)**

16.363 Debug Device Control ODMA (DBGDEV_CTRL_ODMA_REG)—Offset 8748h

This register contains a number of fields that provide a specific level of configurability for the OUT DMA that is part of Debug Device logic. This configurability is above and beyond that defined in the xHCI specification.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RW	**Reserved (RSVD)**
18	0h RW	**Enable ACK FIFO credit accounting (EN_ACK_FCA):** Setting this field will enable ACK FIFO credit accounting. ODMA will ensure that ample room exists in the ACK FIFO for expected device responses prior to initiating a given DP
17:14	0h RW	**Reserved (RSVD_1)**
13	0h RW	**Enable ACK FIFO ICA mechanisms (EN_ACK_FIFO_ICA):** Setting this field will enable ACK FIFO individual credit accounting mechanisms for Async vs. Periodic Endpoints. ODMA will ensure that ample room exists in the ACK FIFO for expected device responses prior to initiating a given DP
12:9	0h RW	**Reserved (RSVD_2)**

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW	**Clear ownership of context semaphore (CL_OWN_CS):** Setting this field generates a pulse that clears the ownership of the context semaphore that is shared between the Out DMA Response and Completion Finite State Machines
7	0h RW	**Return OD ACK credits (RET_OD_ACK_CR):** Setting this field generates a pulse that implicitly returns all of the Out DMA ACK credits on all ports
6	0h RW	**Reserved (RSVD_3)**
5	0h RW	**Return ODCF SM to idle state (RET_ODCF_SM_IS):** Setting this field generates a pulse that returns the Out DMA Completion Finite State Machine into the IDLE state
4	0h RW	**Return ODRF SM to idle state (RET_ODRF_SM_IS):** Setting this field generates a pulse that returns the Out DMA Response Finite State Machine into the IDLE state
3	0h RW	**Return ODRDF SM to idle state (RET_ODRDF_SM_IS):** Setting this field generates a pulse that returns the Out DMA Read Finite State Machine into the IDLE state
2:0	0h RW	**Reserved (RSVD_4)**

16.364 DBC Control Register 1 (DBCCTL_REG)—Offset 8760h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD):** RSVD
7	0h RW	**SW_DCE_SEL (SW_DCE_SEL)**
6:3	0h RW	**DISC_RXD_CNT (DISC_RXD_CNT)**
2	0h RO	Reserved.
1	0h RW	**Force DCE Mode (FORCE_DCE_MODE):** 0: When DCE is set, the DbC switches to Mode 2 1: When DCE is set, the DbC switches to Mode 3

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**Force Disconnect upon DCE (FORCE_DISCONNECT_ON_DCE):** If this bit is set by BIOS, the DbC will temporarily disconnect from the remote host if the DCE is set, and shortly thereafter re-connect. This allows the DbC to switch from Mode1 to Mode2 or Mode 3 operation upon DCE being set.

16.365 SSIC Policy and Implementation Specific Registers Capability ID register (SSIC_IMPLEMENTATION_CAPABILITY_ID_REG)—Offset 8800h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 40C4h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD)**
15:8	40h RO	**Next Capability Pointer (NCP)**
7:0	C4h RO	**Supported Protocol ID (PID)**

16.366 SSIC global configuration control register (SSIC_GLOBAL_CONFIG_CONTROL_REG)—Offset 8804h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Reserved (RSVD)**
17:16	0h RW	**T_ACT_H8_MARGIN (T_ACT_H8_MARGIN):** Specifies the margin time added to the tActivate spec time of 100us in T_ACT_H8_TIME. 11: Reserved 10: 50us 01: 20us (default for ModPHY) 00: 10us (default for MIPI MPHY)
15:4	0h RO	**Reserved1 (RSVD1)**

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**BB_PLL_OVRD_DURING_PWM (BB_PLL_OVRD_DURING_PWM)**
2	0h RW	**CMN_LANE_PWRGATE_DIS (CMN_LANE_PWRGATE_DIS):** This will disable the MPHY common lane power gate.
1	0h RW	**HS_CLK_GATE_DIS (HS_CLK_GATE_DIS):** This will disable the HS Clock gate request from XHCI to MIPI PLL.
0	0h RW	**PWM_CLK_GATE_DIS (PWM_CLK_GATE_DIS):** This will disable the PWM Clock gate request from XHCI to MIPI PLL.

16.367 (PORT1_SSIC_CONFIG_REG1)—Offset 8808h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 82286010h

Bit Range	Default & Access	Field Name (ID): Description
31:30	2h RW	**NUM_RRAP_ATTEMPT (NUM_RRAP_ATTEMPT):** Specify the number of RRAP access attempt if there was tRRAPInitiatorResponse timeout.
29:22	8h RW	**TX_MIN_STALL (TX_MIN_STALL):** (TX_Min_STALL_NoConfig_Time_Capability in MIPI PHY Spec). PA will ensure TX_MIN_STALL time in STALL before entering HS-BURST or HIBERN8. Specifies minimum time in SI in STALL state.
21:17	14h RW	**PWM_EXIT_TIME (PWM_EXIT_TIME):** This will be max of RX_Min_ActivateTime_Capability and TX_Min_SAVE_Config_Time_Capability in MIPI PHY Specs) Specifies minimum activate time needed in 5us steps. PA will wait PWM_EXIT_TIME time after exiting PWM to start HS-BURST. SSIC spec requires it to be at 100us. For speedup mode this time would be in 1us steps.
16	0h RW	**DSP_DISC_BURST_CLOSE_RRAP (DSP_DISC_BURST_CLOSE_RRAP):** 0 - Send BURST Closure RRAP for DSP Disconnect 1 - Dont send Burst Closure RRAP for DSP Disconnect. For device PWM exit would be indication for DSP Disconnect.
15:13	3h RW	**TX_MIN_ACTIVATE (TX_MIN_ACTIVATE):** (TX_Min_ActivateTime in MIPI PHY Spec) Specifies minimum activate time needed in 100us steps. PA will wait TX_MIN_ACTIVATE time after H8 exit to start of PWM Burst. SSIC spec requires it to be at 1.5ms. For speedup mode this time would be 10us to 150us in steps of 10us.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**RRAP_BYPASS (RRAP_BYPASS):** Enables DFx Loopback test mode where RRAP command /response is bypassed for HS_CONFIG and PWM_BURST_CLOSURE. Host will still perform initial enter and exit PWM mode with no RRAP transfers.
11	0h RW	**Reserved (RSVD)**
10:8	0h RW	**T_ACT_H8_EXIT (T_ACT_H8_EXIT):** Specifies minimum time in 100us steps for T_ACTIVATE_TIME to exit H8. Val: Si Time / Fast Sim Time 000: 110us / 10us 001: 210us / 20us 110: 710us / 70us 111: 5ms / 500us Spec required timing is 100us.
7:5	0h RW	**MIN_HIBERN8_TIME (MIN_HIBERN8_TIME):** Minimum time in HIBERN8 state 0 110us 1 210us .. 7 810us PA will ensure MIN_HIBERN8_TIME in its TX HIBERN8. For speedup mode this time will be from 10us to 80us in steps of 10us.
4	1h RW	**SSICRATE (SSICRATE):** 0 B Series 1 A Series
3:2	0h RW	**HSGEAR (HSGEAR):** 0=HS-G1 1=HS-G2 2=HS-G3
1:0	0h RW	**SSICLANE (SSICLANE):** 0=Single lane 1=Bi-lane 2=Quad-lane

16.368 (PORT1_SSIC_CONFIG_REG2)—Offset 880Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 4881F46h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**SSIC_PORT_UNUSED (SSIC_PORT_UNUSED):** This indicates that no SSIC device is connected to the port and PORTSC register would always reflect DISCONNECTED.
30	0h RW	**PROG_DONE (PROG_DONE):** BIOS will program this bit once SSIC profile programming is done. SSIC will enable the MPHY and local PLL once this bit is set.
29:26	1h RW	**NUM_OF_MK0 (NUM_OF_MK0):** Number of MK0 which transmitter will send before sending any data.
25	0h RW	**DISABLE_SCRAMBLING (DISABLE_SCRAMBLING):** This bit will disable scrambling in HS-BURST
24:21	4h RW	**RETRAIN_TIME (RETRAIN_TIME):** Corresponds to time in 10us to detect improper training of the local and remote M-RX as part of HS-BURST entry. SSIC Spec specific 40 to 50us
20:16	8h RW	**PHY_RESET_TIME (PHY_RESET_TIME):** Corresponds to time in 100ns, PA will drive PHY RESET for MIPI PHY.

Bit Range	Default & Access	Field Name (ID): Description
15:8	1Fh RW	**LRST_TIME (LRST_TIME):** Corresponds to time in 100us PA will drive DIF-P for line reset. MIPI PHY specifies 3.1 ms Minimum. For speedup mode this time will be in steps of 5us.
7:0	46h RW	**ACTIVATE_LRST_TIME (ACTIVATE_LRST_TIME):** (Corresponds to tResetDIFN) Specifies in step of 1ms period of time a DSP is required to drive a DIF-N prior to a LINE-RESET. SSIC spec defined range is 60-80ms. For speedup mode this time will be from 5us to 400us in steps of 5us.

16.369 (PORT1_SSIC_CONFIG_REG3)—Offset 8810h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 40400000h

Bit Range	Default & Access	Field Name (ID): Description
31:28	4h RW	**U0_STALL_TO (U0_STALL_TO):** Time in 2^[U-STALL_TO] clock for U0 STALL entry. If there is no packet in arbiter for U0_STALL_TO time, PA will enable U0 STALL if DISABLE_U0_STALL is not set. 0000 0011 (Not recommended) 0100 16 SSIC HS CLK . 1111 32768 SSIC HS CLK
27	0h RW	**MPHY_TEST_MODE_EN (MPHY_TEST_MODE_EN):** When this bit is set, controller would not initiate any PWM once it enters PWM Mode. It will wait for tester to send Loopback RRAP Command.
26	0h RW	**DL_PWR_GATE_DIS (DL_PWR_GATE_DIS)**
25:21	2h RW	**HIBERN8_ENTER_TX (HIBERN8_ENTER_TX):** The time PA will drive DIF-N after last bit of Line-CFG before entering H8. In steps of 50ns. Legal range is 50 to 1000ns.
20:19	0h RW	**LUP_LDN_TIMER_MAX (LUP_LDN_TIMER_MAX):** 00: 10 us 01: 100 us 10: 1000us /1ms 11: (Timer is disable and dont transmit LDN)
18:3	0h RW	**RESERVED (RSVD):** rsvd
2	0h RW	**DISABLE_U0_STALL (DISABLE_U0_STALL):** This bit will disable the STALL entry in U0
1	0h RW	**SSIC_PG_U3_DIS (SSIC_PG_U3_DIS):** This disables MPHY DL PG during U3.
0	0h RW	**SSIC_PG_U2_DIS (SSIC_PG_U2_DIS):** This disables MPHY DL PG during U2.

16.370 (PORT1_SSIC_CONFIG_REG4)—Offset 8814h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 64796464h

Bit Range	Default & Access	Field Name (ID): Description
31:24	64h RW	**PM_LC_MAX_TIMER (PM_LC_MAX_TIMER):** In steps of 1us. Default 100us
23:16	79h RW	**PM_ENTRY_TIMER_MAX (PM_ENTRY_TIMER_MAX):** In steps of 1us. Default 100us
15:8	64h RW	**PEND_HP_TIMER_MAX (PEND_HP_TIMER_MAX):** In steps of 1us. Default 100us
7:0	64h RW	**CRD_PEND_TIMER_MAX (CRD_PEND_TIMER_MAX):** In steps of 1us. Default 100us

16.371 (PORT1_SSIC_LOOPBACK_CONFIG_REG)—Offset 8818h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**RESERVED (RSVD)**
7:4	0h RW	**LOOPBACK_EN (LOOPBACK_EN):** This Register is used to enable Conformance Loopback in the Target in the specified PAIR.. Bit [0]: Writing 1b1 enables the loopback mode in PAIR0. Writing 1b0 shall have no effect. Bit [1]: Writing 1b1 enables the loopback mode in PAIR1. Writing 1b0 shall have no effect. Bit [2]: Writing 1b1 enables the loopback mode in PAIR2. Writing 1b0 shall have no effect. Bit [3]: Writing 1b1 enables the loopback mode in PAIR3. Writing 1b0 shall have no effect.
3:0	0h RW	**RX LOOPBACK CNTR RESET (RX_LOOPBACK_CNTR_RESET):** Reset RX_BURST_COUNT and RX_ERR_COUNT, write only, self-clearing. Bit [0] for PAIR0 Bit [1] for PAIR1 Bit [2] for PAIR2 Bit [3] for PAIR3

16.372 (PORT1_SSIC_LOOPBACK_BURST_COUNT_REG)—Offset 881Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**RX_BURST_COUNT_LANE3 (RX_BURST_COUNT_LANE3):** RX_BURST_COUNT shall increment each time the M-RX of the PAIR under test detects a prepare time DIF-P. This shall be reset when the DUT receives a Write Command with rx_count_reset set to one. These counters shall not be reset when the DUT receives a LINE-RESET.
23:16	0h RO	**RX_BURST_COUNT_LANE2 (RX_BURST_COUNT_LANE2):** RX_BURST_COUNT shall increment each time the M-RX of the PAIR under test detects a prepare time DIF-P. This shall be reset when the DUT receives a Write Command with rx_count_reset set to one. These counters shall not be reset when the DUT receives a LINE-RESET.
15:8	0h RO	**RX_BURST_COUNT_LANE1 (RX_BURST_COUNT_LANE1):** RX_BURST_COUNT shall increment each time the M-RX of the PAIR under test detects a prepare time DIF-P. This shall be reset when the DUT receives a Write Command with rx_count_reset set to one. These counters shall not be reset when the DUT receives a LINE-RESET.
7:0	0h RO	**RX_BURST_COUNT_LANE0 (RX_BURST_COUNT_LANE0):** RX_BURST_COUNT shall increment each time the M-RX of the PAIR under test detects a prepare time DIF-P. This shall be reset when the DUT receives a Write Command with rx_count_reset set to one. These counters shall not be reset when the DUT receives a LINE-RESET.

16.373 (PORT1_SSIC_LOOPBACK_ERROR_COUNT_REG)—Offset 8820h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**RX_ERR_COUNT_LANE3 (RX_ERR_COUNT_LANE3):** RX_BURST_COUNT shall increment each time the M-RX of the PAIR under test detects a prepare time DIF-P. This shall be reset when the DUT receives a Write Command with rx_count_reset set to one. These counters shall not be reset when the DUT receives a LINE-RESET.

Bit Range	Default & Access	Field Name (ID): Description
23:16	0h RO	**RX_ERR_COUNT_LANE2 (RX_ERR_COUNT_LANE2):** RX_BURST_COUNT shall increment each time the M-RX of the PAIR under test detects a prepare time DIF-P. This shall be reset when the DUT receives a Write Command with rx_count_reset set to one. These counters shall not be reset when the DUT receives a LINE-RESET.
15:8	0h RO	**RX_ERR_COUNT_LANE1 (RX_ERR_COUNT_LANE1):** RX_BURST_COUNT shall increment each time the M-RX of the PAIR under test detects a prepare time DIF-P. This shall be reset when the DUT receives a Write Command with rx_count_reset set to one. These counters shall not be reset when the DUT receives a LINE-RESET.
7:0	0h RO	**RX_ERR_COUNT_LANE0 (RX_ERR_COUNT_LANE0):** RX_BURST_COUNT shall increment each time the M-RX of the PAIR under test detects a prepare time DIF-P. This shall be reset when the DUT receives a Write Command with rx_count_reset set to one. These counters shall not be reset when the DUT receives a LINE-RESET.

16.374 SSIC Local and Remote Profile Registers Capability ID register (SSIC_PROFILE_CAPABILITY_ID_REG)—Offset 8900h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: FFC5h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD)**
15:8	FFh RO	**Next Capability Pointer (NCP)**
7:0	C5h RO	**Supported Protocol ID (PID)**

16.375 SSIC Port N Register Access Control (PORT1_REGISTER_ACCESS_CONTROL)—Offset 8904h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: C00000h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (RSVD)**
25	0h RW	**RRAP Register Bank Valid (RRAP_REG_BANK_VALID):** 0 = No valid Commands in the Lane N register bank. Host Controller can close the PWM burst once commands from this register are completed. 1 = Valid commands present in the Lane N register bank. Host controller must complete commands from the register bank before closing the burst.
24	0h RW	**Target Phy (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy Setting this bit to 1 allows the use of this command mechanism to write to local Phy profile and AFE tuning registers Primarily as a back up option.
23	1h RW	**HS_Config (HS_CONFIG):** When this bit is set to 1 the host controller will issue an RRAP write with HS_Config=1 once it sees Command Phase Done = 1
22	1h RW	**Command Phase Done (CPD):** When set to 1, this indicates that SW has completed performing RRAP cycles through the command register.
21	0h RW	**Command Valid (CMD_VALID):** When written to 1 indicates that the Attribute ID and Attribute Data for writes fields are valid.
20	0h RW	**Read_Write (READ_WRITE):** 0 = Write 1= Read
19:8	0h RW	**Attribute ID (ATT_ID):** Attribute ID that is being written or read
7:0	0h RW	**Attribute Write Data (ATT_WRITE_DATA):** Data byte that is required to be written to either the local phy or the remote phy

16.376 SSIC Port N Register Access Status (PORT1_REGISTER_ACCESS_STATUS)—Offset 8908h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RO	**Reserved (RSVD)**
9:8	0h RW	**Command Completion Status (CCS):** 00 = Command not complete 01 = Command complete with Success 10 = Command complete with Error These bits must be cleared before a new command is initiated.
7:0	0h RO	**Read data (READ_DATA):** Data read as a result of the RRAP operation

16.377 (PORT1_PROFILE_ATTRIBUTES_REG0)—Offset 890Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.378 (PORT1_PROFILE_ATTRIBUTES_REG1)—Offset 8910h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.379 (PORT1_PROFILE_ATTRIBUTES_REG2)—Offset 8914h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.380 (PORT1_PROFILE_ATTRIBUTES_REG3)—Offset 8918h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.381 (PORT1_PROFILE_ATTRIBUTES_REG4)—Offset 891Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.382 (PORT1_PROFILE_ATTRIBUTES_REG5)—Offset 8920h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.383 (PORT1_PROFILE_ATTRIBUTES_REG6)—Offset 8924h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.384 (PORT1_PROFILE_ATTRIBUTES_REG7)—Offset 8928h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.385 (PORT1_PROFILE_ATTRIBUTES_REG8)—Offset 892Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.386 (PORT1_PROFILE_ATTRIBUTES_REG9)—Offset 8930h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.387 (PORT1_PROFILE_ATTRIBUTES_REG10)—Offset 8934h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.388 (PORT1_PROFILE_ATTRIBUTES_REG11)—Offset 8938h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.389 (PORT1_PROFILE_ATTRIBUTES_REG12)—Offset 893Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.390 (PORT1_PROFILE_ATTRIBUTES_REG13)—Offset 8940h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.391 (PORT1_PROFILE_ATTRIBUTES_REG14)—Offset 8944h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.392 (PORT1_PROFILE_ATTRIBUTES_REG15)—Offset 8948h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.393 (PORT1_PROFILE_ATTRIBUTES_REG16)—Offset 894Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.394 (PORT1_PROFILE_ATTRIBUTES_REG17)—Offset 8950h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.395 (PORT1_PROFILE_ATTRIBUTES_REG18)—Offset 8954h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.396 (PORT1_PROFILE_ATTRIBUTES_REG19)—Offset 8958h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.397 (PORT1_PROFILE_ATTRIBUTES_REG20)—Offset 895Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.398 (PORT1_PROFILE_ATTRIBUTES_REG21)—Offset 8960h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.399 (PORT1_PROFILE_ATTRIBUTES_REG22)—Offset 8964h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.400 (PORT1_PROFILE_ATTRIBUTES_REG23)—Offset 8968h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.401 (PORT1_PROFILE_ATTRIBUTES_REG24)—Offset 896Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.402 (PORT1_PROFILE_ATTRIBUTES_REG25)—Offset 8970h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.403 (PORT1_PROFILE_ATTRIBUTES_REG26)—Offset 8974h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.404 (PORT1_PROFILE_ATTRIBUTES_REG27)—Offset 8978h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.405 (PORT1_PROFILE_ATTRIBUTES_REG28)—Offset 897Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.406 (PORT1_PROFILE_ATTRIBUTES_REG29)—Offset 8980h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.407 (PORT1_PROFILE_ATTRIBUTES_REG30)—Offset 8984h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.408 (PORT1_PROFILE_ATTRIBUTES_REG31)—Offset 8988h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.409 (PORT1_PROFILE_ATTRIBUTES_REG32)—Offset 898Ch

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.410 (PORT1_PROFILE_ATTRIBUTES_REG33)—Offset 8990h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.411 (PORT1_PROFILE_ATTRIBUTES_REG34)—Offset 8994h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.412 (PORT1_PROFILE_ATTRIBUTES_REG35)—Offset 8998h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.413 (PORT1_PROFILE_ATTRIBUTES_REG36)—Offset 899Ch

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.414 (PORT1_PROFILE_ATTRIBUTES_REG37)—Offset 89A0h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.415 (PORT1_PROFILE_ATTRIBUTES_REG38)—Offset 89A4h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.416 (PORT1_PROFILE_ATTRIBUTES_REG39)—Offset 89A8h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.417 (PORT1_PROFILE_ATTRIBUTES_REG40)—Offset 89ACh

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.418 (PORT1_PROFILE_ATTRIBUTES_REG41)—Offset 89B0h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.419 (PORT1_PROFILE_ATTRIBUTES_REG42)—Offset 89B4h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.420 (PORT1_PROFILE_ATTRIBUTES_REG43)—Offset 89B8h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.421 (PORT1_PROFILE_ATTRIBUTES_REG44)—Offset 89BCh

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.422 (PORT1_PROFILE_ATTRIBUTES_REG45)—Offset 89C0h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.423 (PORT1_PROFILE_ATTRIBUTES_REG46)—Offset 89C4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.424 (PORT1_PROFILE_ATTRIBUTES_REG47)—Offset 89C8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.425 (PORT1_PROFILE_ATTRIBUTES_REG48)—Offset 89CCh

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.426 (PORT1_PROFILE_ATTRIBUTES_REG49)—Offset 89D0h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.427 (PORT1_PROFILE_ATTRIBUTES_REG50)—Offset 89D4h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.428 (PORT1_PROFILE_ATTRIBUTES_REG51)—Offset 89D8h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.429 (PORT1_PROFILE_ATTRIBUTES_REG52)—Offset 89DCh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.430 (PORT1_PROFILE_ATTRIBUTES_REG53)—Offset 89E0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.431 (PORT1_PROFILE_ATTRIBUTES_REG54)—Offset 89E4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.432 (PORT1_PROFILE_ATTRIBUTES_REG55)—Offset 89E8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.433 (PORT1_PROFILE_ATTRIBUTES_REG56)—Offset 89ECh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.434 (PORT1_PROFILE_ATTRIBUTES_REG57)—Offset 89F0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.435 (PORT1_PROFILE_ATTRIBUTES_REG58)—Offset 89F4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.436 (PORT1_PROFILE_ATTRIBUTES_REG59)—Offset 89F8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.437 (PORT1_PROFILE_ATTRIBUTES_REG60)—Offset 89FCh

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.438 (PORT1_PROFILE_ATTRIBUTES_REG61)—Offset 8A00h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.439 (PORT1_PROFILE_ATTRIBUTES_REG62)—Offset 8A04h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.440 (PORT1_PROFILE_ATTRIBUTES_REG63)—Offset 8A08h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.441 GLOBAL_TIME_SYNC_CAP_REG (GLOBAL_TIME_SYNC_CAP_REG)—Offset 8E10h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 12C9h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED (RSVD)**
15:8	12h RO	**Next Capability pointer (NCP)**
7:0	C9h RO	**Capability ID (CID)**

16.442 GLOBAL_TIME_SYNC_CTRL_REG (GLOBAL_TIME_SYNC_CTRL_REG)—Offset 8E14h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	**RESERVED (RSVD)**
0	0h RW/1S	**Time Stamp Counter Capture Initiate (TIME_STAMP_CNTR_CAPTURE_INITIATE):** SW sets this bit to initiate a time capture. Once the time capture is complete and the time values are valid in the Local and Global time capture registers, HW clears the bit.

16.443 MICROFRAME_TIME_REG (MICROFRAME_TIME_REG)—Offset 8E18h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**RESERVED (RSVD)**
29:16	0h RO	**Captured Frame List Current Index/Frame Number (CMFI):** The value in this register is updated in response to sample_now signal. Bits [29:16] reflect state of bits [13:0] of FRINDEX
15:13	0h RO	**RESERVED1 (RSVD1)**
12:0	0h RO	**Captured Micro-frame BLIF (CMFB):** The value is updated in response to sample_now signal and provides information about offset within micro-frame. Captured value represents number of 8 high-speed bit time units from start of micro-frame. At the beginning of micro-frame captured value will be 0 and increase to maximum value at the end. Default maximum value is 7499 but it may be changed as result of adjustment done via Bus Interval Adjust (BIA).

16.444 GLOBAL_TIME_LOW_REG (GLOBAL_TIME_LOW_REG)—Offset 8E20h

Global Time Value (Low):

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**GLOBAL_TIME_LOW_REG (GLOBAL_TIME_LOW):** Global Time Value (Low):

16.445 GLOBAL_TIME_HI_REG (GLOBAL_TIME_HI_REG)—Offset 8E24h

Global Time Value (High):

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**GLOBAL_TIME_HI_REG (GLOBAL_TIME_HI):** Global Time Value (High):

16.446 Debug Status Capability Register (DEBUG_STATUS_CAPABILITY_REG)—Offset 8E58h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: CBh

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd (Rsvd)**
15:8	0h RO	**Next Capability Pointer (NCP)**
7:0	CBh RO	**Capability ID (CID)**

16.447 Host Ctrl USB3 Soft Error Count Register 1 (HOST_CTRL_USB3_ERR_COUNT_REG1)—Offset 8E5Ch

This register is updated by hardware and cleared by software

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED (cfg_reserved)**
15:0	0h RO	**USB3_SOFT_ERR_CNT (cfg_usb3_soft_error_cnt)**

16.448 Host Ctrl USB3 Soft Error Count Register 2 (HOST_CTRL_USB3_ERR_COUNT_REG2)—Offset 8E60h

This register is updated by hardware and cleared by software

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED (cfg_reserved)**
15:0	0h RO	**USB3_SOFT_ERR_CNT (cfg_usb3_soft_error_cnt)**

16.449 Host Ctrl USB3 Soft Error Count Register 3 (HOST_CTRL_USB3_ERR_COUNT_REG3)—Offset 8E64h

This register is updated by hardware and cleared by software

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED (cfg_reserved)**
15:0	0h RO	**USB3_SOFT_ERR_CNT (cfg_usb3_soft_error_cnt)**

16.450 Host Ctrl USB3 Soft Error Count Register 4 (HOST_CTRL_USB3_ERR_COUNT_REG4)—Offset 8E68h

This register is updated by hardware and cleared by software

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED (cfg_reserved)**
15:0	0h RO	**USB3_SOFT_ERR_CNT (cfg_usb3_soft_error_cnt)**

16.451 Host Ctrl USB3 Soft Error Count Register 5 (HOST_CTRL_USB3_ERR_COUNT_REG5)—Offset 8E6Ch

This register is updated by hardware and cleared by software

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED (cfg_reserved)**
15:0	0h RO	**USB3_SOFT_ERR_CNT (cfg_usb3_soft_error_cnt)**

16.452 Host Ctrl USB3 Soft Error Count Register 6 (HOST_CTRL_USB3_ERR_COUNT_REG6)—Offset 8E70h

This register is updated by hardware and cleared by software

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED (cfg_reserved)**
15:0	0h RO	**USB3_SOFT_ERR_CNT (cfg_usb3_soft_error_cnt)**

16.453 Host Ctrl USB3 Soft Error Count Register 7 (HOST_CTRL_USB3_ERR_COUNT_REG7)—Offset 8E74h

This register is updated by hardware and cleared by software

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED (cfg_reserved)**
15:0	0h RO	**USB3_SOFT_ERR_CNT (cfg_usb3_soft_error_cnt)**

16.454 DBC_GP2_OUT_PAYLOAD_BP_LOW (DBC_GP2_OUT_PAYLOAD_BP_LOW)—Offset 0h

Lower DW of Payload Address Pointer

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DBC_GP2_OUT_PAYLOAD_BP_LOW (DBC_GP2_OUT_PAYLOAD_BP_LOW):** Lower DW of Payload Address Pointer

16.455 DBC_GP2_OUT_PAYLOAD_BP_HI (DBC_GP2_OUT_PAYLOAD_BP_HI)—Offset 4h

Upper DW of Payload Address Pointer

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DBC_GP2_OUT_PAYLOAD_BP_HI (DBC_GP2_OUT_PAYLOAD_BP_HI):** Upper DW of Payload Address Pointer

16.456 DBC_GP2_OUT_PAYLOAD_QUALIFIERS (DBC_GP2_OUT_PAYLOAD_QUALIFIERS)—Offset 8h

DbC GP2 OUT Payload Qualifiers

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (RSVD)**
21:8	0h RW	**DESTN_ID (DESTN_ID):** Decode_Mode[1:0] PSF[3:0] Port_Group[0:0] Port[3:0] Channel[2:0]
7:4	0h RO	**RESERVED1 (RSVD1)**
3:0	0h RW	**ROOT_SPACE (ROOT_SPACE)**

16.457 DBC_GP2_OUT_PAYLOAD_TRANSFER_LENGTH (DBC_GP2_OUT_PAYLOAD_TRANSFER_LENGTH)—Offset Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**DATA_VALID (DATA_VALID):** This bit indicates the presence of a valid data buffer and is the doorbell that allows the DbC to initiate a DMA transaction. This bit is cleared by HW upon completion of the DMA upto either the specific transfer length, or thehost controller sending a short packet
30:0	0h RW	**PAYLOAD_TRANSFER_LENGTH (PAYLOAD_TRANSFER_LENGTH):** Length of payload buffer in Bytes (1 based count)

16.458 DBC_GP2_OUT_STATUS_BP_LOW (DBC_GP2_OUT_STATUS_BP_LOW)—Offset 10h

Lower DW of STATUS Address Pointer

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DBC_GP2_OUT_STATUS_BP_LOW (DBC_GP2_OUT_STATUS_BP_LOW):** Lower DW of STATUS Address Pointer

16.459 DBC_GP2_OUT_STATUS_BP_HI (DBC_GP2_OUT_STATUS_BP_HI)—Offset 14h

Upper DW of STATUS Address Pointer

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DBC_GP2_OUT_STATUS_BP_HI (DBC_GP2_OUT_STATUS_BP_HI):** Upper DW of STATUS Address Pointer

16.460 DBC_GP2_OUT_STATUS_QUALIFIERS (DBC_GP2_OUT_STATUS_QUALIFIERS)—Offset 18h

DbC GP2 OUT STATUS Qualifiers

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (RSVD)**
21:8	0h RW	**DESTN_ID (DESTN_ID):** Decode_Mode[1:0] PSF[3:0] Port_Group[0:0] Port[3:0] Channel[2:0]
7:4	0h RO	**RESERVED1 (RSVD1)**
3:0	0h RW	**ROOT_SPACE (ROOT_SPACE)**

16.461 DBC_GP2_IN_PAYLOAD_BP_LOW (DBC_GP2_IN_PAYLOAD_BP_LOW)—Offset 1Ch

Lower DW of Payload Address Pointer

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DBC_GP2_IN_PAYLOAD_BP_LOW (DBC_GP2_IN_PAYLOAD_BP_LOW):** Lower DW of Payload Address Pointer

16.462 DBC_GP2_IN_PAYLOAD_BP_HI (DBC_GP2_IN_PAYLOAD_BP_HI)—Offset 20h

Upper DW of Payload Address Pointer

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DBC_GP2_IN_PAYLOAD_BP_HI (DBC_GP2_IN_PAYLOAD_BP_HI):** Upper DW of Payload Address Pointer

16.463 DBC_GP2_IN_PAYLOAD_QUALIFIERS (DBC_GP2_IN_PAYLOAD_QUALIFIERS)—Offset 24h

DbC GP2 OUT Payload Qualifiers

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (RSVD)**

Bit Range	Default & Access	Field Name (ID): Description
21:8	0h RW	**DESTN_ID (DESTN_ID):** Decode_Mode[1:0] PSF[3:0] Port_Group[0:0] Port[3:0] Channel[2:0]
7:4	0h RO	**RESERVED1 (RSVD1)**
3:0	0h RW	**ROOT_SPACE (ROOT_SPACE)**

16.464 DBC_GP2_IN_PAYLOAD_TRANSFER_LENGTH (DBC_GP2_IN_PAYLOAD_TRANSFER_LENGTH)—Offset 28h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**DATA_VALID (DATA_VALID):** This bit indicates the presence of a valid data buffer and is the doorbell that allows the DbC to initiate a DMA transaction. This bit is cleared by HW upon completion of the DMA upto either the specific transfer length, or thehost controller sending a short packet
30:0	0h RW	**PAYLOAD_TRANSFER_LENGTH (PAYLOAD_TRANSFER_LENGTH):** Length of payload buffer in Bytes (1 based count)

16.465 DBC_GP2_IN_STATUS_BP_LOW (DBC_GP2_IN_STATUS_BP_LOW)—Offset 2Ch

Lower DW of STATUS Address Pointer

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DBC_GP2_IN_STATUS_BP_LOW (DBC_GP2_IN_STATUS_BP_LOW):** Lower DW of STATUS Address Pointer

16.466 DBC_GP2_IN_STATUS_BP_HI (DBC_GP2_IN_STATUS_BP_HI)—Offset 30h

Upper DW of STATUS Address Pointer

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DBC_GP2_IN_STATUS_BP_HI (DBC_GP2_IN_STATUS_BP_HI):** Upper DW of STATUS Address Pointer

16.467 DBC_GP2_IN_STATUS_QUALIFIERS (DBC_GP2_IN_STATUS_QUALIFIERS)—Offset 34h

DbC GP2 OUT STATUS Qualifiers

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (RSVD)**
21:8	0h RW	**DESTN_ID (DESTN_ID):** Decode_Mode[1:0] PSF[3:0] Port_Group[0:0] Port[3:0] Channel[2:0]
7:4	0h RO	**RESERVED1 (RSVD1)**
3:0	0h RW	**ROOT_SPACE (ROOT_SPACE)**

16.468 DBC_DFX_OUT_CONTROL (DBC_DFX_OUT_CONTROL)—Offset 38h

DBC_DFX_OUT_CONTROL

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**RESERVED (RSVD)**
8	0h RW	**Credit De-allocate Update (CREDIT_DE_ALLOC_UPDATE):** This is sent by ExI Bridge to DbC whenever a packet is dispatched from the Inbound Buffer or in response to credit Init request. 1: 1 credit dispatched due to deallocation 0: Absolute value of credits in response to a Credit Init request from DbC
7:5	0h RO	**RESERVED (RSVD1)**
4:0	0h RW	**AVAILABLE_CREDITS (AVAILABLE_CREDITS):** ExI Bridge writes the absolute value of the available credit count into this register each time it drains an entry from its buffer. Each credit represents a 64 Byte ExI Packet. Also, coming out of reset, DbC will request for a Credit Init, and ExI Bridge will write the absolute value of available credit to this register.

16.469 DBC_DFX_IN_PAYLOAD_BP_LOW (DBC_DFX_IN_PAYLOAD_BP_LOW)—Offset 3Ch

Lower DW of Payload Address Pointer

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DBC_DFX_IN_PAYLOAD_BP_LOW (DBC_DFX_IN_PAYLOAD_BP_LOW):** Lower DW of Payload Address Pointer

16.470 DBC_DFX_IN_PAYLOAD_BP_HI (DBC_DFX_IN_PAYLOAD_BP_HI)—Offset 40h

Upper DW of Payload Address Pointer

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DBC_DFX_IN_PAYLOAD_BP_HI (DBC_DFX_IN_PAYLOAD_BP_HI):** Upper DW of Payload Address Pointer

16.471 DBC_DFX_IN_PAYLOAD_TRANSFER_LENGTH (DBC_DFX_IN_PAYLOAD_TRANSFER_LENGTH)—Offset 44h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**DATA_VALID (DATA_VALID):** This bit indicates the presence of a valid data buffer and is the doorbell that allows the DbC to initiate a DMA transaction. This bit is cleared by HW upon completion of the DMA upto either the specific transfer length, or thehost controller sending a short packet
30:0	0h RW	**PAYLOAD_TRANSFER_LENGTH (PAYLOAD_TRANSFER_LENGTH):** Length of payload buffer in Bytes (1 based count)

16.472 DBC_DFX_IN_STATUS_BP_LOW (DBC_DFX_IN_STATUS_BP_LOW)—Offset 48h

Lower DW of STATUS Address Pointer

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DBC_DFX_IN_STATUS_BP_LOW (DBC_DFX_IN_STATUS_BP_LOW):** Lower DW of STATUS Address Pointer

16.473 DBC_DFX_IN_STATUS_BP_HI (DBC_DFX_IN_STATUS_BP_HI)—Offset 4Ch

Upper DW of STATUS Address Pointer

Access Method

Type: MEM Register (Size: 32 bits)	**Device:** **Function:**

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DBC_DFX_IN_STATUS_BP_HI (DBC_DFX_IN_STATUS_BP_HI):** Upper DW of STATUS Address Pointer

16.474 DBC_TRACE_IN_PAYLOAD_BP_LOW (DBC_TRACE_IN_PAYLOAD_BP_LOW)—Offset 50h

Lower DW of Payload Address Pointer

Access Method

Type: MEM Register (Size: 32 bits)	**Device:** **Function:**

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DBC_TRACE_IN_PAYLOAD_BP_LOW (DBC_TRACE_IN_PAYLOAD_BP_LOW):** Lower DW of Payload Address Pointer

16.475 DBC_TRACE_IN_PAYLOAD_BP_HI (DBC_TRACE_IN_PAYLOAD_BP_HI)—Offset 54h

Upper DW of Payload Address Pointer

Access Method

Type: MEM Register (Size: 32 bits)	**Device:** **Function:**

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DBC_TRACE_IN_PAYLOAD_BP_HI (DBC_TRACE_IN_PAYLOAD_BP_HI):** Upper DW of Payload Address Pointer

16.476 DBC_TRACE_IN_PAYLOAD_QUALIFIERS (DBC_TRACE_IN_PAYLOAD_QUALIFIERS)—Offset 58h

DbC TRACE OUT Payload Qualifiers

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (RSVD)**
21:8	0h RW	**DESTN_ID (DESTN_ID):** Decode_Mode[1:0] PSF[3:0] Port_Group[0:0] Port[3:0] Channel[2:0]
7:4	0h RO	**RESERVED1 (RSVD1)**
3:0	0h RW	**ROOT_SPACE (ROOT_SPACE)**

16.477 DBC_TRACE_IN_PAYLOAD_TRASNFER_DOORBELL (DBC_TRACE_IN_PAYLOAD_TRANSFER_DOORBELL)—Offset 5Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**RSVD (RSVD)**
18:11	0h RW	**PAYLOAD_OFFSET (PAYLOAD_OFFSET):** The Offset in multiples of 1024B from the base pointer from which to transfer the payload referenced by this doorbell.
10:0	0h RW	**PAYLOAD_TRASNFER_LENGTH (PAYLOAD_TRASNFER_LENGTH):** Length of payload buffer in Bytes (1 based count)

16.478 DBC_TRACE_IN_STATUS_BP_LOW (DBC_TRACE_IN_STATUS_BP_LOW)—Offset 60h

Lower DW of STATUS Address Pointer

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DBC_TRACE_IN_STATUS_BP_LOW (DBC_TRACE_IN_STATUS_BP_LOW):** Lower DW of STATUS Address Pointer

16.479 DBC_TRACE_IN_STATUS_BP_HI (DBC_TRACE_IN_STATUS_BP_HI)—Offset 64h

Upper DW of STATUS Address Pointer

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DBC_TRACE_IN_STATUS_BP_HI (DBC_TRACE_IN_STATUS_BP_HI):** Upper DW of STATUS Address Pointer

16.480 DBC_TRACE_IN_STATUS_QUALIFIERS (DBC_TRACE_IN_STATUS_QUALIFIERS)—Offset 68h

DbC TRACE OUT STATUS Qualifiers

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (RSVD)**
21:8	0h RW	**DESTN_ID (DESTN_ID):** Decode_Mode[1:0] PSF[3:0] Port_Group[0:0] Port[3:0] Channel[2:0]
7:4	0h RO	**RESERVED1 (RSVD1)**
3:0	0h RW	**ROOT_SPACE (ROOT_SPACE)**

16.481 DBC_ERROR_CONTROL_STATUS_REG (DBC_ERROR_CONTROL_STATUS_REG)—Offset 6Ch

DBC_ERROR_CONTROL_STATUS_REG

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Reserved (RSVD)**
8	0h RW	**DBC_TRACE_IN (DBC_TRACE_IN)**
7	0h RW	**DBC_DFX_OUT (DBC_DFX_OUT)**
6	0h RW	**DBC_DFX_IN (DBC_DFX_IN)**
5	0h RW	**DBC_GP2_OUT (DBC_GP2_OUT)**
4	0h RW	**DBC_GP2_IN (DBC_GP2_IN)**
3	0h RW	**DBC_GP1_OUT (DBC_GP1_OUT)**
2	0h RW	**DBC_GP1_IN (DBC_GP1_IN)**
1	0h RW	**CONTROL_EP (CONTROL_EP)**
0	0h RW	**STATUS_WRITE_ON_ERROR_EN (STATUS_WRITE_ON_ERROR_EN):** Enable status write generation on detecting error on Trace/DFx/GP2 EPs

16.482 DBC_EXI_CONTROL_STATUS_REG (DBC_EXI_CONTROL_STATUS_REG)—Offset 70h

DBC_EXI_CONTROL_STATUS_REG

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**DBC_GP2_IN_STATUS_LOG (DBC_GP2_IN_STATUS_LOG):** Set when the DMA transfer request status for GP2 IN is completed. Cleared when a new DMA transfer request is received.
30	0h RO	**DBC_GP2_OUT_STATUS_LOG (DBC_GP2_OUT_STATUS_LOG):** Set when the DMA transfer request status for GP2 OUT is completed. Cleared when a new DMA transfer request is received.
29:13	0h RO	**RSVD (RSVD)**
12	0h RW	**DBC_EXI_CLK_GATING_CTRL (DBC_EXI_CLK_GATING_CTRL):** Disable gating for prim_clk for DbC-ExI logic. By default clock gating is enabled.
11	0h RW	**DBC_SAI_CHECK_POLICY (DBC_SAI_CHECK_POLICY):** 1: Disable SAI checking on read requests only 0: Enable Default SAI check (on Reads & Writes)
10	0h RW	**TRACE_SEQUENCER_PIPELINE_DISABLE (TRACE_SEQUENCER_PIPELINE_DISABLE):** Set to disable pipelining of read requests to fabric for different trace transfer requests. By default, trace sequencer would start issuing reads requests to Fabric for a subsequent transfer request while the reads for the previous one hasnt been returned back to DbC 0: Read Pipelining enabled for Trace 1: Disable read pipelining for Trace
9:7	0h RW	**DBC_MAX_PKT_SIZE (DBC_MAX_PKT_SIZE):** Configurable MPS for Trace, DFx, GP2 EPs Default = 1KB 000 : 1KB 001 : 64B 010 : 128B 011 : 256B 100 : 512B Others : Reserved
6:1	0h RW	**DBC_NUM_PENDING_READS (DBC_NUM_PENDING_READS):** Constrain number of read requests pending on Fabric on behalf of DbC Eps. Sequencers limit the number of read requests that will be initiated on Fabric at any given time to NUM_PENDING_READ. NUM_PENDING_READ = [1..READS_PER_MPS] Where READS_PER_MPS is the number of read requests require for 1 MPS for the given EP. Applies for DbC.[Trace|DFx|GP2] IN EPs
0	0h RW	**DBC_TRACE_PIPELINE_DEPTH1 (DBC_TRACE_PIPELINE_DEPTH1):** Constrain DbC.Trace IN Pipeine to 1 deep. Limit processing of DbC Trace to one transfer at any time. In other wordsUntil Trace data for the first transfer request from NPKH is completed on USB3 interface and status returned back to NPKH, subsequent transfer request is not started, only 1 Trace sequencer is active at any given time.

16.483 FABRIC_AGENT_UPSTREAM_GRANT_COUNT_REG (FABRIC_AGENT_UPSTREAM_GRANT_COUNT_REG)—Offset 74h

FABRIC_AGENT_UPSTREAM_GRANT_COUNT_REG

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 9249249h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	Reserved.
29:27	1h RW	**IN_PORT_ARB_GRANT_COUNT (IN_PORT_ARB_GRANT_COUNT)**
26:24	1h RW	**GP2_IN_PORT_ARB_GRANT_COUNT (GP2_IN_PORT_ARB_GRANT_COUNT)**
23:21	1h RW	**DFX_IN_PORT_ARB_GRANT_COUNT (DFX_IN_PORT_ARB_GRANT_COUNT)**
20:18	1h RW	**TRACE_IN_PORT_ARB_GRANT_COUNT (TRACE_IN_PORT_ARB_GRANT_COUNT)**
17:15	1h RW	**RD_REQ_HYSTERESIS_COUNT (RD_REQ_HYSTERESIS_COUNT)**
14:12	1h RW	**DBC_GRANT_COUNT (DBC_GRANT_COUNT)**
11:9	1h RW	**EXI_GRANT_COUNT (EXI_GRANT_COUNT)**
8:6	1h RW	**GP2_GRANT_COUNT (GP2_GRANT_COUNT)**
5:3	1h RW	**DFX_GRANT_COUNT (DFX_GRANT_COUNT)**
2:0	1h RW	**TRACE_GRANT_COUNT (TRACE_GRANT_COUNT)**

16.484 DBC_ECO_POLICY_REG1 (DBC_ECO_POLICY_REG1)—Offset 78h

ECO Policy registers for future use

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 40F00h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**RESERVED (RSVD)**
23	0h RW	**GP2_IN_BACK2BACK_SHORT_PKT_FIX_EN (GP2_IN_BACK2BACK_SHORT_PKT_FIX_EN)**

Bit Range	Default & Access	Field Name (ID): Description
22	0h RW	**DFX_IN_BACK2BACK_SHORT_PKT_FIX_EN (DFX_IN_BACK2BACK_SHORT_PKT_FIX_EN)**
21	0h RW	**TRACE_IN_BACK2BACK_SHORT_PKT_FIX_EN (TRACE_IN_BACK2BACK_SHORT_PKT_FIX_EN)**
20:18	1h RW	**OUT_PORT_ARB_GRANT_COUNT (OUT_PORT_ARB_GRANT_COUNT)**
17	0h RO	Reserved.
16	0h RW	**OUT_EP_BURST_SIZE (OUT_EP_BURST_SIZE):** Default: HW selects the burst size for OUT EPs based on the mode of operation Kernel Debug Mode Only: Supported OUT EP = 1 (GP1) ; Burst Size = 4 Platform Debug Mode Only: Supported OUT EPs = 2 (DFx, GP2), Burst Size = 2 Kernel and Platform debug mode: Supported OUT EPs = 3 (GP1, DFx, GP2) ; Burst Size = 1 When Set to 1 ; Burst Size = 1
15	0h RW	**IGNORE_EXI_EN (IGNORE_EXI_EN)**
14	0h RW	**IGNORE_DCE (IGNORE_DCE)**
13	0h RW	**DBC_EXI_HCRESET_DISABLE (DBC_EXI_HCRESET_DISABLE):** When set to 1, DBC/DBC-EXI will not take any action on HCReset. Default behavior is to trigger HCReset flow.
12	0h RW	**DBC_ALLOW_PG (DBC_ALLOW_PG):** When set to 1, DBC/DBC-EXI will allow PG under HW initiated low power modes (D0ix). This could be used in a flow where XHC is allowed to enter low power modes, while debug data will subsequently transported to debug host after re-entry into
11:8	Fh RW	**HC_RESET_WPR_HANDLING_TIMEOUT_VALUES (HC_RESET_WPR_HANDLING_TIMEOUT_VALUES):** 0000 : 0 us (trigger exit right away) 0001 : 250 us 0010 : 500 us 1111 : 3.75 ms
7	0h RW	**TIMEOUT_BASED_HC_RESET_WPR_HANDLING (TIMEOUT_BASED_HC_RESET_WPR_HANDLING):** When set, enable timer based mechanism to allow HCReset and Warm Port Reset to override any hung situation due to any flows not completing as expected. Use HC Reset/WPR Handling Timeout values for forcing HCReset/ WPR flows
6	0h RW	**RSVD1 (RSVD1):** Reserved
5	0h RW	**INTERNAL_DBC_EXI_ENABLE (INTERNAL_DBC_EXI_ENABLE):** Enable address decode of transactions using Source Decode transactions based on Source ID instead of the default MBAR based address based decode.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW	**EXI_ENABLE_OVERRIDE_ENABLE (EXI_ENABLE_OVERRIDE_ENABLE):** When this bit is set, the Internal DBC ExI Enable takes priority over the external ExI Enable signal received from EXI Bridge IP Default behavior is to use exio_scr_ectrl_eavails input from ExI Bridge
3	0h RW	**TRACE_STATUS_ADDRESS (TRACE_STATUS_ADDRESS):** When this bit is set, the address for Status Write request will include the address offset. Default address for Trace Status is the base address of the trace request
2	0h RW	**ENABLE_NON_TRANSFER_STATUS (ENABLE_NON_TRANSFER_STATUS):** Enables generation of a status write on each of the interfaces with status indicating the cause of an exception flow trigger condition Default behavior is for the sequencer to return a status completion (for transfer success, or exception scenarios) only when a pending/active DMA is in progress
1	0h RW	**ENABLE_DBC_EXI_ON_CREDIT_INIT_COMPLETION (ENABLE_DBC_EXI_ON_CREDIT_INIT_COMPLETION):** Setting this bit will enforce Credit Init completion as necessary condition to allow transfers on all 3 interfaces (Trace, GP2, DFx). Default behavior requires successful Credit Init completion as necessary condition only to allow DFX OUT EP transfer and not affect other interfaces (Trace, GP2)
0	0h RW	**CREDIT_INIT_TRIGGER (CREDIT_INIT_TRIGGER):** Writing a 1 to this bit will trigger Credit Init request. A Credit Init request will be initiated irrespective of the status of the last Credit Init request initiated.

16.485 DBC_ECO_POLICY_REG2 (DBC_ECO_POLICY_REG2)—Offset 7Ch

ECO Policy registers for future use

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**RESERVED (RSVD)**
5	0h RO	**DBC_EXI_RUN (DBC_EXI_RUN):** Internal version of RUN. Set only when DBC_EXI is enabled and link up as upstream port was due to DBC_EXI =1

Bit Range	Default & Access	Field Name (ID): Description
4	0h RO	**DBC_RUN (DBC_RUN):** Internal version of RUN bit. Set only when DBC(DD) is enabled and link up as upstream port was due to DCE=1
3	0h RO	**DBC_EXI_EN (DBC_EXI_EN):** DBC_EXI_EN Internal version of EXI_EN
2	0h RO	**DBC_EN (DBC_EN):** DBC_EN Internal version of DBC_EN tracked by HW
1:0	0h RO	**CREDIT_INIT_STATUS (CREDIT_INIT_STATUS):** Tracks the status of Credit Init Handshake. HW updates this register. 00 : Credit Init not yet requested 01 : Credit Init Pending: request sent to ExI Bridge Not received Init Credit value from ExI Bridge 10 : Credit Init Done : Received Init Credit value from ExI Bridge 11 : Reserved

16.486 DBC_ECO_POLICY_REG3 (DBC_ECO_POLICY_REG3)—Offset 80h

ECO Policy registers for future use

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DBC_ECO_POLICY_REG3 (DBC_ECO_POLICY_REG3)**

16.487 DBC_ECO_POLICY_REG4 (DBC_ECO_POLICY_REG4)—Offset 84h

ECO Policy registers for future use

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DBC_ECO_POLICY_REG4 (DBC_ECO_POLICY_REG4)**

16.488 DBConEXI Capability Port Status and Control Register (DBC_EXI_DCPORTSC)—Offset 88h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 80h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD)**
23	0h RW/C	**Port Config Error Change (CEC):** This flag indicates that the port failed to configure its link partner. 0 = No change. 1 = Port Config Error detected. Software shall clear this bit by writing a '1' to it.
22	0h RW/C	**Port Link Status Change (PLC):** This flag is set to '1' due to the following PLS transitions: U0 -) U3 Suspend signaling detected from Debug Host U3 -) U0 Resume complete Polling -) Disabled Training Error Ux or Recovery -) Inactive Error Software shall clear this bit by writing a '1' to it. This field is '0' if DCE is '0'
21	0h RW/C	**Port Reset Change (PRC):** This bit is set when reset processing on this port is complete (i.e. a '1' to '0' transition of PR). '0' = No change. '1' = Reset complete.Software shall clear this bit by writing a '1' to it. This field is '0' if DCE is '0'.
20:18	0h RO	**Reserved (RSVD_1)**
17	0h RW/C	**Connect Status Change (CSC):** '1' = Change in Current Connect Status. '0' = No change. Indicates a change has occurred in the port's Current Connect Status. The xHC sets this bit to '1' for all changes to the Debug Device connect status, even if system software has not cleared an existing DbC Connect Status Change. For example, the insertion status changes twice before system software has cleared the changed condition, hardware will be "setting" an already-set bit (i.e., the bit will remain '1'). Software shall clear this bit by writing a '1' to it. This field is '0' if DCE is '0'.
16:14	0h RO	**Reserved (RSVD_2)**
13:10	0h RO	**Port Speed (PSPD):** This field identifies the speed of the port. This field is only relevant when a Debug Host is attached (CCS = '1') in all other cases this field shall indicate Undefined Speed. 0 Undefined Speed 1-15 Protocol Speed ID (PSI) Note: The Debug Capability onlydoes not supports LS, FS, or HS operation.
9	0h RO	**Reserved (RSVD_3)**

Bit Range	Default & Access	Field Name (ID): Description
8:5	4h RO	**Port Link State (PLS):** This field reflects its current link state. This field is only relevant when a Debug Host is attached (Debug Port Number) '0'). Value Meaning 0 Link is in the U0 State 1 Link is in the U1 State 2 Link is in the U2 State 3 Link is in the U3 State (Device Suspended) 4 Link is in the Disabled State 5 Link is in the RxDetect State 6 Link is in the Inactive State 7 Link is in the Polling State 8 Link is in the Recovery State 9 Link is in the Hot Reset State 15:10 Reserved Note: Transitions between different states are not reflected until the transition is complete.
4	0h RO	**Port Reset (PR):** '1' = Port is in Reset. '0' = Port is not in Reset. This bit is set to '1' when the bus reset sequence as defined in the USB Specification is detected on the Root Hub port assigned to the Debug capability. It is cleared when the bus reset sequence is completed by the Debug Host, and the DbC shall transition to the USB Default state. A '0' to '1' transition of this bit shall clear DBC_EXI_DCPORTSC PED ('0'). This field is '0' if DCE or CCS are '0'.
3:2	0h RO	**Reserved (RSVD_4)**
1	0h RW	**Port Enabled/Disabled (PED):** Default = '0'. '1' = Enabled. '0' = Disabled. This flag shall be set to '1' by a '0' to '1' transition of CCS or a '1' to '0' transition of the PR. When PED transitions from '1' to '0' due to the assertion of PR, the port's link shall transition to the Rx.Detect state. This flag may be used by software to enable or disable the operation of the Root Hub port assigned to the Debug Capability. The Debug Capability Root Hub port operation may be disabled by a fault condition (disconnect event or other fault condition, e.g. a LTSSM Polling substate timeout, tPortConfiguration timeout error, etc.), the assertion of DBC_EXI_DCPORTSC PR, or by software. 0 = Debug Capability Root Hub port is disabled. 1 = Debug Capability Root Hub port is enabled. When the port is disabled (PED = '0') the port's link shall enter the SS.Disabled state and remain there until PED is reasserted ('1') or DCE is negated ('0'). Note that the Root Hub port is remains mapped to Debug Capability while PED = '0'. While PED = '0' the Debug Capability will appear to be disconnected to the Debug Host. Note, this bit is not affected by PORTSC PR bit transitions. This field is '0' if DCE or CCS are '0'.
0	0h RO	**Current Connect Status (CCS):** '1' = A Root Hub port is connected to a Debug Host and assigned to the Debug Capability. '0' = No Debug Host is present. This value reflects the current state of the port, and may not correspond to the value reported by the Connect Status Change (CSC) field in the Port Status Change Event that was generated by a '0' to '1' transition of this bit. This flag is '0' if Debug Capability Enable (DCE) is '0'.

16.489 DBC_GP2_OUT_DMA_STS_REG1 (DBC_GP2_OUT_DMA_STS_REG1)—Offset 8Ch

GP2 OUT DMA Status Register1

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RW	**Reserved (RSVD)**
4	0h RW	**DBC_OUT_DMA_STS_VALID (DBC_OUT_DMA_STS_VALID):** GP2 OUT DMA status valid
3:0	0h RW	**DBC_GP2_OUT_DMA_STS (DBC_GP2_OUT_DMA_STS):** Status of GP2 OUT DMA transfer

16.490 DBC_GP2_OUT_DMA_STS_REG2 (DBC_GP2_OUT_DMA_STS_REG2)—Offset 90h

GP2 OUT DMA Status Register2

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD1):** Status of GP2 OUT DMA transfer
30:0	0h RW	**DBC_GP2_OUT_DMA_DATA_LEN (DBC_GP2_OUT_DMA_DATA_LEN):** GP2 OUT DMA Data length in bytes

16.491 DBC_GP2_IN_DMA_STS_REG1 (DBC_GP2_IN_DMA_STS_REG1)—Offset 94h

GP2 IN DMA Status Register1

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RW	**Reserved (RSVD)**
4	0h RW	**DBC_GP2_IN_DMA_STS_VALID (DBC_GP2_IN_DMA_STS_VALID):** GP2 OUT DMA status valid
3:0	0h RW	**DBC_GP2_IN_DMA_STS (DBC_GP2_IN_DMA_STS):** Status of GP2 IN DMA transfer

16.492 DBC_GP2_IN_DMA_STS_REG2 (DBC_GP2_IN_DMA_STS_REG2)—Offset 98h

GP2 IN DMA Status Register2

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD1):** Status of GP2 IN DMA transfer
30:0	0h RW	**DBC_GP2_IN_DMA_DATA_LEN (DBC_GP2_IN_DMA_DATA_LEN):** GP2 IN DMA Data length in bytes

16.493 DEBUG_SW_CONTROL_STATUS_REG (DEBUG_SW_CONTROL_STATUS_REG)—Offset 100h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**RSVD (RSVD)**
7	0h RW	**FAST_SW_TIMEOUT_MODE (FAST_SW_TIMEOUT_MODE):** 1 : Enable Fast Timeout Simulation Mode. When set, timeout values used for SW response Timeout mode will be in micro-seconds. 0 : Normal mode. Timeout values are as defined by SW response Timeout register.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW/1C	**SW_TIMEOUT_STATUS (SW_TIMEOUT_STATUS):** 1: Set by HW when Timeout occurred and HW disabled SW Handling 0: Timeout not triggered Set by HW, Cleared by SW. Timeout will also clear Debug SW Available bit.
5:2	0h RW	**SW_RESPONSE_TIMEOUT (SW_RESPONSE_TIMEOUT):** 0000 : 1 ms 0001: 20 ms 0010: 40 ms 0011: 60 ms 0100: 80 ms 0101: 100 ms 0110: 200ms 1111: 1100ms
1	0h RW	**SW_RESPONSE_TIMEOUT_ENABLE (SW_RESPONSE_TIMEOUT_ENABLE):** 1: HW will timeout and trigger STALL response if SW did not respond within SW Timeout period 0: Timeout is not triggered.
0	0h RW	**DEBUG_SW_AVAILABLE (DEBUG_SW_AVAILABLE):** 1: Debug SW has been loaded and is available 0: Debug SW is not available When Debug SW is available, then HW presents the Control Transfer requests to SW and relies on SW to handle the requests.

16.494 DEBUG_REQUEST_INFO_AND_STATUS_REG (DEBUG_REQUEST_INFO_AND_STATUS_REG)—Offset 104h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**RESPONSE_PACKET_LAST (RESPONSE_PACKET_LAST):** Specifies that this response packet is the last packet for satisfying the current request or if there is additional packets to complete the request. 1 : Current response packet is the last data packet for the current request 0: Current response packet is not the last data packet for the current request. There is one or more additional data packets required to complete the request. For example, If the response data buffer is 512B, and the response is > 512B, then there will be multiple response data packets required, and this bit is set to 0 except for the last one.
30:27	0h RW	**RESPONSE_TYPE (RESPONSE_TYPE):** Specifies the type of response that needs to be generated 000 : DATA Response 001: ACK 010: STALL (Request not supported) Others: Reserved For DATA response, associate DATA is written into the Response Stack by SW.

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW/1S	**DEBUG_RESPONSE_AVAIL (DEBUG_RESPONSE_AVAIL):** 1: Valid SW response available for HW 0: No response available SW sets this bit when it has generated a response for a debug request received. HW clears this bit after the response has been consumed (completed the transfer to host).
25:15	0h RO	**RSVD (RSVD)**
14:12	0h RW	**DEBUG_REQUEST_ERROR_TYPE (DEBUG_REQUEST_ERROR_TYPE):** Indicates that cause/Type of the error detect. Valid only when Debug Request Error Detected bit is 1 Encoding
11	0h RW	**DEBUG_REQUEST_ERROR_DETECTED (DEBUG_REQUEST_ERROR_DETECTED):** HW sets this bit to indicate to Debug SW that an error was detected on the packet received on the USB3 link
10:1	0h RO	**DEBUG_REQUEST_LENGTH (DEBUG_REQUEST_LENGTH):** Length of any data associated with the current request. This field is valid only when a debug request is pending.
0	0h RW/1C	**DEBUG_REQUEST_PENDING (DEBUG_REQUEST_PENDING):** 1: Debug request is available for SW to consume 0: No request is pending for SW HW sets this bit to indicate to Debug SW that a control transfer request is pending for SW to handle. SW clears this bit once it has consumed the request.

16.495 DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_0)—Offset 108h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_WORD):** 64B Stack for incoming Device request and Data

16.496 DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_1)—Offset 10Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_WORD):** 64B Stack for incoming Device request and Data

16.497 DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_2)—Offset 110h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_WORD):** 64B Stack for incoming Device request and Data

16.498 DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_3)—Offset 114h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_WORD):** 64B Stack for incoming Device request and Data

16.499 DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_4)—Offset 118h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_WORD):** 64B Stack for incoming Device request and Data

16.500 DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_5)—Offset 11Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_WORD):** 64B Stack for incoming Device request and Data

16.501 DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_6)—Offset 120h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_WORD):** 64B Stack for incoming Device request and Data

16.502 DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_7)—Offset 124h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_WORD):** 64B Stack for incoming Device request and Data

16.503 DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_8)—Offset 128h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_WORD):** 64B Stack for incoming Device request and Data

16.504 DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_9)—Offset 12Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_WORD):** 64B Stack for incoming Device request and Data

16.505 DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_10)—Offset 130h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_WORD):** 64B Stack for incoming Device request and Data

16.506 DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_11)—Offset 134h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_WORD):** 64B Stack for incoming Device request and Data

16.507 DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_12)—Offset 138h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_WORD):** 64B Stack for incoming Device request and Data

16.508 DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_13)—Offset 13Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_WORD):** 64B Stack for incoming Device request and Data

16.509 DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_14)—Offset 140h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_WORD):** 64B Stack for incoming Device request and Data

16.510 DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_REG_15)—Offset 144h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_REQUEST_STACK (DEBUG_REQUEST_STACK_WORD):** 64B Stack for incoming Device request and Data

16.511 DEBUG_RESPONSE_INFO_AND_STATUS_REG (DEBUG_RESPONSE_INFO_AND_STATUS_REG)—Offset 148h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**RESPONSE_DATA_LENGTH (RESPONSE_DATA_LENGTH):** Length of data associated with the current response. This field is valid only when a debug response is available. (bit[0] is set). The length of the data packet will be equal to the response data stack size except when the Request Packet Last = 1 where it may a short packet (< response data stack size)
15:0	0h RO	**RSVD (RSVD)**

16.512 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_0)—Offset 180h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.513 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_1)—Offset 184h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.514 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_2)—Offset 188h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.515 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_3)—Offset 18Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.516 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_4)—Offset 190h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.517 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_5)—Offset 194h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.518 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_6)—Offset 198h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.519 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_7)—Offset 19Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.520 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_8)—Offset 1A0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.521 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_9)—Offset 1A4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.522 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_10)—Offset 1A8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.523 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_11)—Offset 1ACh

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.524 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_12)—Offset 1B0h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.525 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_13)—Offset 1B4h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.526 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_14)—Offset 1B8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.527 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_15)—Offset 1BCh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.528 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_16)—Offset 1C0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.529 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_17)—Offset 1C4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.530 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_18)—Offset 1C8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.531 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_19)—Offset 1CCh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.532 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_20)—Offset 1D0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.533 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_21)—Offset 1D4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.534 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_22)—Offset 1D8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.535 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_23)—Offset 1DCh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.536 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_24)—Offset 1E0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.537 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_25)—Offset 1E4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.538 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_26)—Offset 1E8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.539 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_27)—Offset 1ECh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.540 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_28)—Offset 1F0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.541 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_29)—Offset 1F4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.542 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_30)—Offset 1F8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.543 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_31)—Offset 1FCh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.544 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_32)—Offset 200h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.545 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_33)—Offset 204h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.546 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_34)—Offset 208h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.547 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_35)—Offset 20Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.548 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_36)—Offset 210h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.549 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_37)—Offset 214h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.550 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_38)—Offset 218h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.551 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_39)—Offset 21Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.552 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_40)—Offset 220h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.553 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_41)—Offset 224h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.554 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_42)—Offset 228h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.555 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_43)—Offset 22Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.556 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_44)—Offset 230h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.557 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_45)—Offset 234h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.558 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_46)—Offset 238h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.559 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_47)—Offset 23Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.560 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_48)—Offset 240h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.561 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_49)—Offset 244h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.562 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_50)—Offset 248h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.563 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_51)—Offset 24Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.564 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_52)—Offset 250h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.565 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_53)—Offset 254h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.566 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_54)—Offset 258h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.567 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_55)—Offset 25Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.568 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_56)—Offset 260h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.569 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_57)—Offset 264h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.570 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_58)—Offset 268h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.571 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_59)—Offset 26Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.572 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_60)—Offset 270h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.573 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_61)—Offset 274h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.574 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_62)—Offset 278h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.575 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_63)—Offset 27Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.576 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_64)—Offset 280h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.577 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_65)—Offset 284h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.578 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_66)—Offset 288h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.579 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_67)—Offset 28Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.580 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_68)—Offset 290h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.581 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_69)—Offset 294h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.582 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_70)—Offset 298h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.583 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_71)—Offset 29Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.584 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_72)—Offset 2A0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.585 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_73)—Offset 2A4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.586 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_74)—Offset 2A8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.587 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_75)—Offset 2ACh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.588 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_76)—Offset 2B0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.589 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_77)—Offset 2B4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.590 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_78)—Offset 2B8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.591 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_79)—Offset 2BCh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.592 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_80)—Offset 2C0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.593 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_81)—Offset 2C4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.594 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_82)—Offset 2C8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.595 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_83)—Offset 2CCh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.596 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_84)—Offset 2D0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.597 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_85)—Offset 2D4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.598 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_86)—Offset 2D8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.599 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_87)—Offset 2DCh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.600 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_88)—Offset 2E0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.601 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_89)—Offset 2E4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.602 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_90)—Offset 2E8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.603 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_91)—Offset 2ECh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.604 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_92)—Offset 2F0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.605 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_93)—Offset 2F4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.606 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_94)—Offset 2F8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.607 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_95)—Offset 2FCh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.608 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_96)—Offset 300h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.609 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_97)—Offset 304h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.610 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_98)—Offset 308h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.611 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_99)—Offset 30Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.612 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_100)—Offset 310h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.613 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_101)—Offset 314h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.614 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_102)—Offset 318h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.615 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_103)—Offset 31Ch

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.616 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_104)—Offset 320h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.617 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_105)—Offset 324h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.618 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_106)—Offset 328h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.619 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_107)—Offset 32Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.620 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_108)—Offset 330h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.621 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_109)—Offset 334h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.622 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_110)—Offset 338h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.623 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_111)—Offset 33Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.624 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_112)—Offset 340h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.625 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_113)—Offset 344h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.626 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_114)—Offset 348h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.627 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_115)—Offset 34Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.628 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_116)—Offset 350h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.629 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_117)—Offset 354h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.630 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_118)—Offset 358h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.631 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_119)—Offset 35Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.632 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_120)—Offset 360h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.633 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_121)—Offset 364h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.634 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_122)—Offset 368h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.635 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_123)—Offset 36Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.636 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_124)—Offset 370h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.637 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_125)—Offset 374h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.638 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_126)—Offset 378h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.639 DEBUG_RESPONSE_DATA_STACK (DEBUG_RESPONSE_DATA_STACK_REG_127)—Offset 37Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_DATA (DEBUG_RESPONSE_DATA):** Response Data generated by SW for the current response. Data is valid only when a debug response is available (bit[0] is set) and Response Type = DATA

16.640 DEBUG_RESPONSE_HEADER_STACK_REG (DEBUG_RESPONSE_HEADER_STACK_REG_0)—Offset 380h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_HEADER (DEBUG_RESPONSE_HEADER):** Response header generated by SW as a response to the current request/command processed by the SW

16.641 DEBUG_RESPONSE_HEADER_STACK_REG (DEBUG_RESPONSE_HEADER_STACK_REG_1)—Offset 384h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_HEADER (DEBUG_RESPONSE_HEADER):** Response header generated by SW as a response to the current request/command processed by the SW

16.642 DEBUG_RESPONSE_HEADER_STACK_REG (DEBUG_RESPONSE_HEADER_STACK_REG_2)—Offset 388h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DEBUG_RESPONSE_HEADER (DEBUG_RESPONSE_HEADER):** Response header generated by SW as a response to the current request/command processed by the SW

16.643 Vendor ID (VID)—Offset 0h

Access Method

Type: CFG Register
(Size: 16 bits)

Device: 21
Function: 0

Default: 8086h

Bit Range	Default & Access	Field Name (ID): Description
15:0	8086h RO	**Vendor ID (VID)**

16.644 Device ID (DID)—Offset 2h

Access Method

Type: CFG Register
(Size: 16 bits)

Device: 21
Function: 0

Default: 8C31h

Bit Range	Default & Access	Field Name (ID): Description
15:0	8C31h RO/V	**Device ID (DID):** See Global Device ID table in Chap. 6 for value

16.645 Command (CMD)—Offset 4h

Access Method

Type: CFG Register
(Size: 16 bits)

Device: 21
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:11	0h RO	**Reserved (RSVD)**

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW	**Interrupt Disable (ID):** When cleared to 0, the function is capable of generating interrupts. When 1, the function can not generate its interrupt to the interrupt controller. Note that the corresponding Interrupt Status bit is not affected by the interrupt enable.
9	0h RO	**Fast Back to Back Enable (FBE)**
8	0h RW	**SERR# Enable (SERR):** When set to 1, the XHC is capable of generating (internally) SERR#.
7	0h RO	**Wait Cycle Control (WCC)**
6	0h RW	**Parity Error Response (PER):** When set to 1, the XHCI Host Controller will check for correct parity (on its internal interface) and halt operation when bad parity is detected during the data phase as recommended by the XHCI specification. Note that this applies to both requests and completions from the system interface. This bit must be set in order for the parity errors to generate SERR#.
5	0h RO	**VGA Palette Snoop (VPS)**
4	0h RO	**Memory Write Invalidate (MWI)**
3	0h RO	**Special Cycle Enable (SCE)**
2	0h RW	**Bus Master Enable (BME):** When set, it allows XHC to act as a bus master. When cleared, it disable XHC from initiating transactions on the system bus.
1	0h RW	**Memory Space Enable (MSE):** This bit controls access to the XHC Memory Space registers. If this bit is set, accesses to the XHC registers are enabled. The Base Address register for the XHC should be programmed before this bit is set.
0	0h RO	**I/O Space Enable (IOSE):** Reserved as 0. Read-Only.

16.646 Device Status (STS)—Offset 6h

Access Method

Type: CFG Register
(Size: 16 bits)

Device: 21
Function: 0

Default: 290h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**Detected Parity Error (DPE):** This bit is set by the Intel PCH whenever a parity error is seen on the internal interface to the XHC host controller, regardless of the setting of bit 6 or bit 8 in the Command register or any other conditions. Software clears this bit by writing a 1 to this bit location.
14	0h RW/1C	**Signaled System Error (SSE):** This bit is set by the Intel PCH whenever it signals SERR# (internally). The SERR_EN bit (bit 8 in the Command Register) must be 1 for this bit to be set. Software clears this bit by writing a 1 to this bit location.
13	0h RW/1C	**Received Master-Abort Status (RMA):** This bit is set when XHC, as a master, receives a master-abort status on a memory access. This is treated as a Host Error and halts the DMA engines. Software clears this bit by writing a 1 to this bit location.
12	0h RW/1C	**Received Target Abort Status (RTA):** This bit is set when XHC, as a master, receives a target abort status on a memory access. This is treated as a Host Error and halts the DMA engines. Software clears this bit by writing a 1 to this bit location.
11	0h RW/1C	**Signaled Target-Abort Status (STA):** This bit is used to indicate when the XHC function responds to a cycle with a target abort.
10:9	1h RO	**DEVSEL# Timing Status (DEVT):** This 2-bit field defines the timing for DEVSEL# assertion. Read-Only.
8	0h RW/1C	**Master Data Parity Error Detected (MDPED):** This bit is set by the Intel PCH whenever a data parity error is detected on a XHC read completion packet on the internal interface to the XHC host controller and bit 6 of the Command register is set to 1. Software clears this bit by writing a 1 to this bit location.
7	1h RO	**Fast Back-to-Back Capable (FBBC):** Reserved as 1 Read-Only.
6	0h RO	**User Definable Features (UDF):** Reserved as 0. Read-Only.
5	0h RO	**66 MHz Capable (MC):** Reserved as 0. Read-Only.
4	1h RO	**Capabilities List (CL):** Hardwired to 1 indicating that offset 34h contains a valid capabilities pointer.
3	0h RO/V	**Interrupt Status (IS):** This read-only bit reflects the state of this function's interrupt at the input of the enable/disable logic. This bit is a 1 when the interrupt is asserted. This bit will be 0 when the interrupt is deasserted. The value reported in this bit is independent of the value in the Interrupt Enable bit.
2:0	0h RO	**Reserved (RSVD)**

16.647 Revision ID (RID)—Offset 8h

Access Method

Type: CFG Register
(Size: 8 bits)

Device: 21
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO/V	**Revision ID (RID):** See Chap 6 for value.

16.648 Programming Interface (PI)—Offset 9h

Access Method

Type: CFG Register
(Size: 8 bits)

Device: 21
Function: 0

Default: 30h

Bit Range	Default & Access	Field Name (ID): Description
7:0	30h RO	**Programming Interface (PI):** A value of 30h indicates that this USB Host Controller conforms to the XHCI specification.

16.649 Sub Class Code (SCC)—Offset Ah

Access Method

Type: CFG Register
(Size: 8 bits)

Device: 21
Function: 0

Default: 3h

Bit Range	Default & Access	Field Name (ID): Description
7:0	3h RO	**Sub Class Code (SCC):** A value of 03h indicates that this is a Universal Serial Bus Host Controller.

16.650 Base Class Code (BCC)—Offset Bh

Access Method

Type: CFG Register
(Size: 8 bits)

Device: 21
Function: 0

Default: Ch

Bit Range	Default & Access	Field Name (ID): Description
7:0	Ch RO	**Base Class Code (BCC):** A value of 0Ch indicates that this is a Serial Bus controller.

16.651 Master Latency Timer (MLT)—Offset Dh

Access Method

Type: CFG Register (Size: 8 bits)
Device: 21
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Master Latency Timer (MLT):** Because the XHC controller is internally implemented with arbitration on an internal interface, it does not need a master latency timer. The bits will be fixed at 0.

16.652 Header Type (HT)—Offset Eh

Access Method

Type: CFG Register (Size: 8 bits)
Device: 21
Function: 0

Default: 80h

Bit Range	Default & Access	Field Name (ID): Description
7	1h RO	**Multi-Function Bit (MFB):** Read only indicating single function device.
6:0	0h RO	**Configuration layout (CL):** Hardwired to 0 to indicate a standard PCI configuration layout.

16.653 Memory Base Address (MBAR)—Offset 10h

Value in this register will be different after the enumeration process.

Access Method

Type: CFG Register (Size: 64 bits)
Device: 21
Function: 0

Default: 4h

Bit Range	Default & Access	Field Name (ID): Description
63:16	0h RW	**Base Address (BA):** Bits (63:16) correspond to memory address signals (63:16), respectively. This gives 64 KB of relocatable memory space aligned to 64 KB boundaries.
15:4	0h RO	**Reserved (RSVD):** Reserved. Read-Only 0, this indicates that this function is requesting an 64KB block of memory.
3	0h RO	**Prefetchable (Prefetchable):** This bit is hardwired to 0 indicating that this range should not be prefetched.
2:1	2h RO	**Type (Type):** If this field is hardwired to 00 it indicates that this range can be mapped anywhere within 32-bit address space. If this field is hardwired to 10 it indicates that this range can be mapped anywhere within 64-bit address space.
0	0h RO	**Resource Type Indicator (RTE):** This bit is hardwired to 0 indicating that the base address field in this register maps to memory space

16.654 USB Subsystem Vendor ID (SSVID)—Offset 2Ch

This register is modified and maintained by BIOS

Access Method

Type: CFG Register (Size: 16 bits)
Device: 21
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW/L	**USB Subsystem Vendor ID (SSVID):** This register, in combination with the USB Subsystem ID register, enables the operating system to distinguish each subsystem from the others.

16.655 USB Subsystem ID (SSID)—Offset 2Eh

This register is modified and maintained by BIOS

Access Method

Type: CFG Register (Size: 16 bits)
Device: 21
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW/L	**USB Subsystem ID (SSID):** BIOS sets the value in this register to identify the Subsystem ID. This register, in combination with the Subsystem Vendor ID register, enables the operating system to distinguish each subsystem from other(s).

16.656 Capabilities Pointer (CAP_PTR)—Offset 34h

Access Method

Type: CFG Register (Size: 8 bits)
Device: 21
Function: 0

Default: 70h

Bit Range	Default & Access	Field Name (ID): Description
7:0	70h RO	**Capabilities Pointer (CAP_PTR):** This register points to the starting offset of the capabilities ranges.

16.657 Interrupt Line (ILINE)—Offset 3Ch

Access Method

Type: CFG Register (Size: 8 bits)
Device: 21
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Interrupt Line (ILINE):** This data is not used by the Intel PCH. It is used as a scratchpad register to communicate to software the interrupt line that the interrupt pin is connected to.

16.658 Interrupt Pin (IPIN)—Offset 3Dh

Access Method

Type: CFG Register (Size: 8 bits)
Device: 21
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW/L	**Interrupt pin (IPIN):** Bits 7:0 reflect the Interrupt Pin assigned to the host controller by the platform (and are hardwired).

16.659 XHC System Bus Configuration 1 (XHCC1)—Offset 40h

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 0

Default: 1FDh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/O	**Access Control (ACCTRL):** This bit is used by BIOS to lock/unlock lockable bits. When set to '1' the write access to bits locked by this bit is disabled (locked state). When set to '0', the write access to bit locked by this bit is enabled (unlocked state). Writable once after platform reset.
30:25	0h RW	**ECO1 (ECO1):** Reserved
24	0h RW	**Master/Target Abort SERR (RMTASERR):** When set, it allows the out-of-band error reporting from the xHCI Controller to be reported as SERR# (if SERR# reporting is enabled) and thus set the STS.SSE bit.
23	0h RW/C	**Unsupported Request Detected (URD):** Set the HW when xHCI Controller received an unsupported request posted cycle. Cleared by SW when the bit is written with value of '1'.
22	0h RW	**Unsupported Request Report Enable (URRE):** When set this bit allows the URD bit to be reported as SERR# (if SERR# reporting is enabled) and thus set the STS.SSE bit.
21:19	0h RW	**Inactivity Initiated L1 Enable (IIL1E):** If programmed to non-zero, it allows L1 power managed to be enabled after the time-out period specified. 000: Disabled 001: 32 bb_cclk 010: 64 bb_cclk 011: 128 bb_cclk 100: 256 bb_cclk 101: 512 bb_cclk 110: 1024 bb_cclk 111: 131072 bb_cclk
18	0h RW	**XHC Initiated L1 Enable (XHCIL1E):** If set, allow the XHC initiated L1 power management to be enabled.
17	0h RW	**D3 Initiated L1 Enable (D3IL1E):** If set, allow PCI device state D3 initiated L1 power management to be enables.
16:12	0h RW	**Periodic Complete Pre Wake Time (PCPWT):** The value programmed in this field determines how far in advance of the start of the next micro-frame the host controller must de-assert the "Periodic Complete" signal . This allows for platform wake time before the next scheduled periodic transaction. The value programmed in this field represents the # of bytes consumed in the current micro-frame which is required to allow for the periodic complete to de-assert. This allows for a programmable time to cause the periodic complete to de-assert prior to the start of the next micro-frame. Register Format: Bits (16:12) represents the # of bytes remaining with a 256B granularity. Periodic Complete will de-assert if the bytes consumed in the current micro-frame is less

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**SW Assisted xHC Idle (SWAXHCI):** This bit being set will indicate xHC idleness (through SW means), which must be a '1' to allow L1 entry, and subsequently allow backbone clock to be gated. This bit is to be set by Intel xHCI driver after checking that the xHCI Controller will stay in idle state for a significant period of time, e.g. all ports disconnected. This bit can be cleared under the following conditions (see SWAXHCI Policy bits in xHC System Bus Configuration 2 register): n SW: SW could write 0 to clear this bit. n HW: HW, under policy control, will clear this bit on an MMIO access to the Host Controller. n HW: HW, under policy control, will clear this bit when HW exits Idle state.
10:8	1h RW	**L23 to Host Reset Acknowledge Wait Count (L23HRAWC):** If programmed to non zero, it allows a wait period after the L23 PHY has shutdown before returning host reset acknowledge to PMC. 000: Disabled 001: 128 bb_cclk 010: 256 bb_cclk 011: 512 bb_cclk 100: 1024 bb_cclk 101: 2048 bb_cclk 110: 4096 bb_cclk 111: 131072 bb_cclk
7:6	3h RW	**Upstream Type Arbiter Grant Count Posted (UTAGCP):** Grant count for IOSF upstream L2 request type arbiter for posted type
5:4	3h RW	**Upstream Type Arbiter Grant Count Non Posted (UDAGCNP):** Grant count for IOSF upstream L2 type arbiter for non-posted type
3:2	3h RW	**Upstream Type Arbiter Grant Count Completion (UDAGCCP) (UDAGCCP):** Grant count for IOSF upstream L2 type arbiter for completion type
1:0	1h RW	**Upstream Device Arbiter Grant Count (UDAGC) (UDAGC):** Grant count for IOSF upstream L1 device arbiter

16.660 XHC System Bus Configuration 2 (XHCC2)—Offset 44h

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 21
Function: 0

Default: 3C000h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**OC Configuration Done (OCCFGDONE):** This bit is used by BIOS to prevent spurious switching during OC configuration. It must be set by BIOS after configuration of the OC mapping bits is complete. Once this bit is set, OC mapping shall not be changed by SW.
30:26	0h RW	**ECO1 (ECO1):** Reserved
25	0h RW	**DMA Request Boundary Crossing Control (DREQBCC):** This bit controls the boundary crossing limit of each Read/Write Request. 0: 4KB 1: 64B

Bit Range	Default & Access	Field Name (ID): Description
24:22	0h RW	**IDMA Read Request Size Control (IDMA_RDREQSZCTRL):** Read Request Size Control: This bit controls the maximum size of each Read Request. 000: 128B 001: 256B 011 - 110: Reserved 111: 64B
21	0h RW	**XHC Upstream Read Relaxed Ordering Enable (XHCUPRDROE):** This policy controls the Relaxed Ordering attribute for upstream reads. 0 - xHC will clear RO for all upstream read requests. 1 - xHC will set RO for all upstream read requests.
20	0h RW	**IOSF Sideband Register Access Disable (IOSFSRAD):** When set, it disables the IOSF sideband interface from accepting any host space register access.
19:14	Fh RW	**Upstream Non-Posted Pre-Allocation (UNPPA):** This field reserves data sizes, in 64 byte chunks, of the downstream completion resource. This value is zero based. 000000 - 111111: Pre-allocate 64 bytes - 4096 bytes If set greater than the default allows over-allocation If set less than default allows under-allocation Only allowed to be programmed when BME = 0 and no outstanding downstream completion
13:12	0h RW	**SW Assisted xHC Idle Policy (SWAXHCIP):** Note: Irrespective of the setting of this field, SW write of 0 to SWAXHCI will clear the bit. 00b (default): xHC HW clears SWAXHCI bit upon: n MMIO access to Host Controller OR n xHC HW exits Idle state 01b: xHC HW does not autonomously clear SWAXHCI bit. The bit could be cleared only by SW. 10b: xHC HW clears SWAXHCI upon MMIO acces to Host Controller. xHC HW exit from Idle state will not clear SWAXHCI. 11b: Reserved
11	0h RW	**MMIO Read After MMIO Write Delay Disable (RAWDD):** This field controls delay on MMIO Read after MMIO Write. 0b (Default): Delay MMIO Read after MMIO Write 1b: Do not delay MMIO Read after MMIO Write Note that this delay applies after the second of the two DW writes in the case where the IOSF Gasket splits a QW write into two single DW writes to the IP.
10	0h RW	**MMIO Write After MMIO Write Delay Enable (WAWDE):** This field controls delay on MMIO Write after previous MMIO Write. 0b (Default): Do not delay MMIO Write after previous MMIO Write 1b: Delay MMIO Write after previous MMIO Write Note that the delay count does not apply on the second of the two DW writes that are generated by IOSF Gasket when it splits a QW write into two. In other words, the second of the two DW writes could happen without any delay with respect to the first DW write. This choice is being made for ease of ECO. The delay count, in this case, will apply after the second of the two DW writes.
9:8	0h RW	**SW Assisted Cx Inhibit (SWACXIHB):** This field controls how the DMI L1 inhibit signal from USB3 to PMC will behave. 00: Never inhibit Cx 01: Inhibit Cx when Isochronous Endpoint is active (PPT Behavior) 10: Inhibit Cx when Periodic Active as defined in 40.4.3.2.1 11: Always inhibit Cx

Bit Range	Default & Access	Field Name (ID): Description
7:6	0h RW	**SW Assisted DMI L1 Inhibit (SWADMIL1IHB):** This field controls how the DMI L1 inhibit signal from USB3 to DMI will behave. 00: Never inhibit DMI L1. 01: Inhibit DMI L1 when Isochronous Endpoint is active (PPT Behavior). 10: Inhibit DMI L1 when Priodic Active as defined in 40.4.3.2.1. 11: Inhibit DMI L1 if XHCC1.SWAXHCI = 0.
5:3	0h RW	**L1 Force P2 Clock Gating Wait Count (L1FP2CGWC):** If programmed to non zero, it allows L1 force P2 gating off the clock to be delayed after the time-out period specified. If wake up event is detected before the time-out, pclk remains alive and trigger L1 exit as though CPU host is causing the wake, 000: Disabled 001: 128 bb_cclk 010: 256 bb_cclk 011: 512 bb_cclk 100: 1024 bb_cclk 101: 2048 bb_cclk 110: 4096 bb_cclk 111: 131072 bb_cclk
2:0	0h RW	**Read Request Size Control (RDREQSZCTRL):** Read Request Size Control: This bit controls the maximum size of each Read Request. 000: 128B 001: 256B 010: 512B 011 - 110: Reserved 111: 64B

16.661 Clock Gating (XHCLKGTEN)—Offset 50h

Access Method

Type: CFG Register (Size: 32 bits)

Device: 21
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**Rsvd2 (Rsvd2):** Reserved
28	0h RW	**Nak'ing USB2.0 EPs for Backbone Clock Gating and PLL Shutdown (NUEFBCGPS):** This field controls whether Naking USB2.0 EPs, once in Naking low priority schedule, should be considered as active for the considerations for backbone clock gating and PLL shutdown or not. 0: Naking USB2.0 EPs are not considered to be active for Backbone Clock Gating and PLL Shutdown evaluation. 1: Naking USB2.0 EPs are considered to be active for Backbone clock gating and PLL shutdown evaluation.
27	0h RW	**SRAM Power Gate Enable (SRAMPGTEN):** This register enables the SRAM Power Gating when PLL shutdown conditions for all clock domains have been met 0 - Disallow SRAM Power Gating. 1 - Allow SRAM Power Gating
26	0h RW	**SS Link PLL Shutdown Enable (SSLSE):** This register enables the SS P3 state to be exposed to PXP PLL Shutdown conditions on behalf of all USB SS Ports ontop of trunk clock gating. 0 - P3 state NOT allowed to result in PXP PLL shutdown. 1- P3 state allowed to result in PXP PLL shutdown

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**USB2 PLL Shutdown Enable (USB2PLLSE):** When set, this bit allows USB2 PLL to be shutdown when HS Link trunk clock is gated, and xHC can tolerate PLL spin up time for subsequent clock request. Note: if USB2 PLL Shutdown Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value.
24	0h RW	**IOSF Sideband Trunk Clock Gating Enable (IOSFSTCGE):** When set, this bit allows the IOSF sideband clock trunk to be gated when idle conditions are met.
23:20	0h RW	**HS Backbone PXP Trunk Clock Gate Enable (HSTCGE):** This register determines the HS Ux state(s) which will be exposed to Backbone PXP trunk gating of core clock. Uy is a state allowed to result in trunk gating when ss_tcg_ux_en(x) is asserted. (0) ==) U0 or deeper (1) ==) NA (no support for U1) (2) ==) U2 (L1) or deeper (3) ==) U3 (L2) or deeper
19:16	0h RW	**SS Backbone PXP Trunk Clock Gate Enable (SSTCGE):** This register determines the SS Ux state(s) which will be exposed to Backbone PXP trunk gating of core clock. Uy is a state allowed to result in trunk gating when ss_tcg_ux_en(x) is asserted. (0) ==) U0 or deeper (1) ==) U1 or deeper (2) ==) U2 or deeper (3) ==) U3 or deeper
15	0h RW	**XHC Ignore_EU3S (XHCIGEU3S):** This register determines if the xHC will use the EU3S as a condition to allow for Frame timer gating. 0 - xHC may allow frame timer to be gated when EU3S is set and all ports are in the required state. 1 - xHC may allow frame timer to be gated regardless of EU3S.
14	0h RW	**XHC Frame Timer Clock Shutdown Enable (XHCFTCLKSE):** This register determines if the xHC will allow the frame timer clock to be gated. 0 - xHC will not allow ICC PLL 96MHz output to be shutdown thus keeping the frame timer running. 1 - xHC will allow ICC PLL 96MHz output to be shutdown under specific conditions.
13	0h RW	**XHC Backbone PXP Trunk Clock Gate In Presence of ISOCH EP (XHCBBTCGIPISO):** This register controls the policy on allowing Backbone PXP trunk clock gate in the presence of IDLE ISOCH EP s with active DB. Allows the periodic active to be used in enabling Backbone PXP trunk clock gating of core clock. 0 Trunk gate of core clock is not allowed when ISOCH EP DB is set and ISOCH EP s are idle. 1 Allow trunk gate of core clock when ISOCH EP DB is set and ISOCH EP s are idle.
12	0h RW	**XHC HS Backbone PXP Trunk Clock Gate U2 non RWE (XHCHSTCGU2NRWE):** This register controls the policy on allowing Backbone PXP trunk gating of core clock when there is atleast 1 non Remote Wake Enabled HS Port in U2. 0 Prevent trunk gate of core clock when a non RWE HS Port is in U2. 1 Allow trunk gate of core clock when a non RWE HS Port is in U2.

Bit Range	Default & Access	Field Name (ID): Description
11:10	0h RW	**XHC USB2 PLL Shutdown Lx Enable (XHCUSB2PLLSDLE):** This register determines the HS Link state(s) which will be exposed to USB2 PLL Shutdown conditions on behalf of all USB2 HS Ports. Ly is a state allowed to result in USB2 PLL Shutdown when en(x) is asserted. (0) ==) L1 or deeper (1) ==) L2 or deeper
9:8	0h RW	**HS Backbone PXP PLL Shutdown Ux Enable (HSUXDMIPLLSE):** This register determines the Ux state(s) which will be exposed to PXP PLL Shutdown conditions. PLL Shutdown is allowed in: 00b Disabled (Link states shall be disabled for DMI PLL shutdown) 01b U0 or conditions for 10b setting. 10b U2 or conditions for 11b setting. 10b U3, Disconnected, Disabled or Powered-Off.
7:5	0h RW	**SS Backbone PXP PLL Shutdown Ux Enable (SSPLLSUE):** This register determines the Ux state(s) which will be exposed to DMI PLL Shutdown conditions. PLL Shutdown is allowed in: 000b Disabled (Link states shall be ignored for DMI PLL shutdown). 001b U0 or conditions for 010b setting 010b U1 or conditions for 011b setting 011b U2 or conditions for 100b setting 100b U3, Disconnected, Disabled or Powered-Off
4	0h RW	**XHC Backbone Local Clock Gating Enable (XHCBLCGE):** When set, this bit allows XHCI Controller IP backbone clock to be locally gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value.
3	0h RW	**HS Link Trunk Clock Gating Enable (HSLTCGE):** When set, this bit allows High Speed Link control's 480 MHz and its 48/60 MHz link clock trunk to be gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value. BIOS need to query fuses to see if HW is enabled.
2	0h RW	**SS Link Trunk Clock Gating Enable (SSLTCGE):** When set, this bit allows the SuperSpeed Link control's 250 MHz and its divided 125 MHz link clock trunk to ge gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value. BIOS need to query fuses to see if HW is enabled.
1	0h RW	**IOSF Backbone Trunk Clock Gating Enable (IOSFBTCGE):** When set, this bit allows the IOSF backbone clock trunk to be gated when idle conditions are met.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**IOSF Gasket Backbone Local Clock Gating Enable (IOSFBLCGE):** When set, this bit allows the IOSF Gasket backbone clock to be locally gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value. BIOS need to query fuses to see if HW is enabled.

16.662 Audio Time Synchronization (AUDSYNC)—Offset 58h

This 32 bit register is used for audio stream synchronization across different devices. Global signal sample_now captures a value in AUDSYNC register.

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Rsvd2 (Rsvd2)**
29:16	0h RO/V	**Captured Frame List Current Index/Frame Number (CMFI):** The value in this register is updated in response to sample_now signal. Bits (29:16) reflect state of bits (13:0) of FRINDEX
15:13	0h RO	**Rsvd1 (Rsvd1)**
12:0	0h RO/V	**Captured Micro-frame BLIF (CMFB):** The value is updated in response to sample_now signal and provides information about offset within micro-frame. Captured value represents number of 8 high-speed bit time units from start of micro-frame. At the beginning of micro-frame captured value will be 0 and increase to maximum value at the end. Default maximum value is 7499 but it may be changed as result of adjustment done in FLA.

16.663 Serial Bus Release Number (SBRN)—Offset 60h

Access Method

Type: CFG Register (Size: 8 bits)
Device: 21
Function: 0

Default: 30h

Bit Range	Default & Access	Field Name (ID): Description
7:0	30h RO	**Serial Bus Release Number (SBRN):** A value of 30h indicates that this controller follows USB release 3.0.

16.664 Frame Length Adjustment (FLADJ)—Offset 61h

This feature is used to adjust any offset from the clock source that generates the clock that drives the SOF counter. When a new value is written into these six bits, the length of the frame is adjusted. Its initial programmed value is system dependent based on the accuracy of hardware USB clock and is initialized by system BIOS. This register should only be modified when the HChalted bit in the USBSTS register is a one. Changing value of this register while the host controller is operating yields undefined results.

Access Method

Type: CFG Register (Size: 8 bits)
Device: 21
Function: 0

Default: 60h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**Reserved (RSVD):** Read-Only. These bits are reserved for future use and should read as "00".
6	1h RO	**No Frame Length Timing Capability (NO_FRAME_LENGTH_TIMING_CAP):** This flag is set to 1 to indicate that the host controller does not support a programmable Frame Length Timing Value field.
5:0	20h RO	**Frame Length Timing Value (FLTV):** SOF micro-frame length) is equal to 59488 + value in this field. The default value is decimal 32 (20h), which gives a SOF cycle time of 60000. Frame Length (number of High Speed bit times) FLADJ Value (decimal) (decimal) 59488 0 (00h) 59504 1 (01h) 59520 2 (02h) ... 59984 31 (1Fh) 60000 32 (20h) ... 60480 62 (3Eh) 60496 63 (3Fh) Each decimal value change to this register corresponds to 16 high-speed bit times. The SOF cycle time (number of SOF counter clock periods to generate a SOF micro-frame length) is equal to 59488 + value in this field. The default value is decimal 32 (20h), which gives a SOF cycle time of 60000. Frame Length (# High Speed bit times) FLADJ Value

16.665 Best Effort Service Latency (BESL)—Offset 62h

Bset Effort Service Latency.

Access Method

Type: CFG Register (Size: 8 bits)
Device: 21
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:4	0h RW/L	**Default Best Effort Service Latency Deep (DBESLD):** Default Best Effort Service Latency (DBESLD) If the value of this field is non-zero, it defines the recommended value for programming the PORTPMSC register BESLD field. This is programmed by BIOS based on platform parameters.
3:0	0h RW/L	**Default Best Effort Service Latency (DBESL):** If the value of this field is non-zero, it defines the recommended value for programming the PORTPMSC register BESL field. This is programmed by BIOS based on platform parameters.

16.666 PCI Power Management Capability ID (PM_CID)—Offset 70h

Access Method

Type: CFG Register (Size: 8 bits)
Device: 21
Function: 0

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
7:0	1h RO	**PCI Power Management Capability ID (PM_CID):** A value of 01h indicates that this is a PCI Power Management capabilities field.

16.667 Next Item Pointer #1 (PM_NEXT)—Offset 71h

This register is modified and maintained by BIOS

Access Method

Type: CFG Register (Size: 8 bits)
Device: 21
Function: 0

Default: 80h

Bit Range	Default & Access	Field Name (ID): Description
7:0	80h RW/L	**Next Item Pointer #1 (PM_NEXT):** This register defaults to 80h, which indicates that the next capability registers begin at configuration offset 80h. This register is writable when the Access Control bit is set to '0'. This allows BIOS to effectively hide the next capability registers, if necessary. This register should only be written during system initialization before the plug-and-play software has enabled any master-initiated traffic. Values of: 80h implies next capability is MSI 00h implies that MSI capability is hidden. Note: This value is never expected to be programmed.

16.668 Power Management Capabilities (PM_CAP)—Offset 72h

Normally, this register is read-only to report capabilities to the power management software. In order to report different power management capabilities depending on the system in which the Intel PCH is used, the write access to this register is controlled by the Access Control bit (ACCTRL). The value written to this register does not affect the hardware other than changing the value returned during a read. This register is modified and maintained by BIOS

Access Method

Type: CFG Register (Size: 16 bits)
Device: 21
Function: 0

Default: C1C2h

Bit Range	Default & Access	Field Name (ID): Description
15:11	18h RW/L	**PME_Support (PME_Support):** This 5-bit field indicates the power states in which the function may assert PME#. The Intel PCH XHC does not support the D1 or D2 states. For all other states, the Intel PCH XHC is capable of generating PME#. Software should never need to modify this field.
10	0h RW/L	**D2_Support (D2_Support):** The D2 state is not supported.
9	0h RW/L	**D1_Support (D1_Support):** The D1 state is not supported.
8:6	7h RW/L	**Aux_Current (Aux_Current):** The Intel PCH XHC reports 375mA maximum Suspend well current required when in the D3cold state. This value can be written by BIOS when a more accurate value is known.
5	0h RW/L	**DSI (DSI):** The Intel PCH reports 0, indicating that no device-specific initialization is required.
4	0h RO	**Reserved (RSVD)**
3	0h RW/L	**PME Clock (PMEClock):** The Intel PCH reports 0, indicating that no PCI clock is required to generate PME#.
2:0	2h RW/L	**Version (Version):** The Intel PCH reports 010, indicating that it complies with Revision 1.1 of the PCI Power Management Specification.

16.669 Power Management Control/Status (PM_CS)—Offset 74h

Access Method

Type: CFG Register (Size: 16 bits)
Device: 21
Function: 0

Default: 8h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**PME_Status (PME_Status):** This bit is set when the Intel PCH XHC would normally assert the PME# signal independent of the state of the PME_En bit. Writing a 1 to this bit will clear it and cause the internal PME to deassert (if enabled). Writing a 0 has no effect. This bit must be explicitly cleared by the operating system each time the operating system is loaded.
14:13	0h RO	**Data_Scale (Data_Scale):** The Intel PCH hardwires these bits to 00 because it does not support the associated Data register.
12:9	0h RO	**Data_Select (Data_Select):** The Intel PCH hardwires these bits to 0000 because it does not support the associated Data register.
8	0h RW	**PME_En (PME_En):** A 1 enables the Intel PCH XHC to generate an internal PME signal when PME_Status is 1. This bit must be explicitly cleared by the operating system each time it is initially loaded.
7:4	0h RO	**Reserved (RSVD)**
3	1h RO	**No Soft Reset (NSR):** No_Soft_Reset - When set ("1"), this bit indicates that devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset. Configuration Context is preserved. Upon transition from the D3hot to the D0 Initialized state, no additional operating system intervention is required to preserve Configuration Context beyond writing the PowerState bits. Transition from D3hot to D0 by a system or bus segment reset will return to the device state D0 Uninitialized with only PME context preserved if PME is supported and enabled.
2	0h RO	**Reserved (RSVD2)**
1:0	0h RW	**PowerState (PowerState):** This 2-bit field is used both to determine the current power state of XHC function and to set a new power state. The definition of the field values are: 00b - D0 state 11b - D3hot state If software attempts to write a value of 10b or 01b in to this field, the write operation must complete normally, however, the data is discarded and no state change occurs. When in the D3hot state, the Intel PCH must not accept accesses to the XHC memory range, but the configuration space must still be accessible.

16.670 Message Signaled Interrupt CID (MSI_CID)—Offset 80h

Access Method

Type: CFG Register
(Size: 8 bits)

Device: 21
Function: 0

Default: 5h

Bit Range	Default & Access	Field Name (ID): Description
7:0	5h RO	**Capability ID (CID):** Indicates that this is an MSI capability

16.671 Next item pointer (MSI_NEXT)—Offset 81h

Access Method

Type: CFG Register (Size: 8 bits)
Device: 21
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW/L	**Next Pointer (NEXT):** Indicates that this is the last item on the capability list

16.672 Message Signaled Interrupt Message Control (MSI_MCTL)—Offset 82h

Access Method

Type: CFG Register (Size: 16 bits)
Device: 21
Function: 0

Default: 86h

Bit Range	Default & Access	Field Name (ID): Description
15:9	0h RO	**Reserved (RSVD)**
8	0h RO	**Per-Vector Masking Capable (PVM):** Specifies whether controller supports MSI per vector masking. Not supported
7	1h RO	**64 Bit Address Capable (C64):** Specifies whether capable of generating 64-bit messages. This device is 64-bit capable.
6:4	0h RW	**Multiple Message Enable (MME):** Indicates the number of messages the controller should assert. This device supports multiple message MSI.
3:1	3h RO	**Multiple Message Capable (MMC):** Indicates the number of messages the controller wishes to assert. This field must be set by HW to reflect the number of Interrupters supported. Encoding number of Vectors requested (number of Interrupters) 000 1 001 2 010 4 011 8 100 16 101 32 110-111 Reserved
0	0h RW	**MSI Enable (MSIE):** If set to 1, MSI is enabled and the traditional interrupt pins are not used to generate interrupts. If cleared to 0, MSI operation is disabled and the traditional interrupt pins are used.

16.673 Message Signaled Interrupt Message Address (MSI_MAD)—Offset 84h

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 21
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**Addr (Addr):** Lower DW of system specified message address, always DWORD aligned
1:0	0h RO	**Reserved (RSVD)**

16.674 Message Signaled Interrupt Upper Address (MSI_MUAD)—Offset 88h

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 21
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Upper Addr (UpperAddr):** Upper DW of system specified message address.

16.675 Message Signaled Interrupt Message Data (MSI_MD)—Offset 8Ch

Access Method

Type: CFG Register
(Size: 16 bits)

Device: 21
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW	**Data (Data):** This 16-bit field is programmed by system software if MSI is enabled. Its content is driven onto the lower word (PCI AD(15:0)) during the data phase of the MSI memory write transaction. The Multiple Message Enable field (bits 6-4 of the Message Control register) defines the number of low order message data bits the function is permitted to modify to generate its system software allocated vectors. For example, a Multiple Message Enable encoding of 010 indicates the function has been allocated four vectors and is permitted to modify message data bits 1 and 0 (a function modifies the lower message data bits to generate the allocated number of vectors). If the Multiple Message Enable field is 000, the function is not permitted to modify the message data.

16.676 Device Idle Capability (DEVIDLE)—Offset 90h

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 0

Default: F0140009h

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RO	**VID (VID)**
27:24	0h RO	**REV (REV)**
23:16	14h RO	**Length (LENGTH):** Indicates that this capability is 16 bytes long.
15:8	0h RO	**Next Capability Pointer (NCP):** This field contains the offset to the next PCI Capability structure or 000h if no other items exist in the linked list of Capabilities.
7:0	9h RO	**Capability ID (CID)**

16.677 Vendor Specific Header (VSHDR)—Offset 94h

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 0

Default: 1400010h

Bit Range	Default & Access	Field Name (ID): Description
31:20	14h RO	**VSEC Length (VSEC_LENGTH):** This field indicates the number of bytes in the entire VSEC structure, including the PCI Extended Capability header, the Vendor- Specific header, and the Vendor-Specific register
19:16	0h RO	**VSEC Rev (VSEC_REV):** This field is a vendor-defined version number that indicates the version of the VSEC structure. Software must qualify the Vendor ID and VSEC ID before interpreting this field.
15:0	10h RO	**VSEC ID (VSEC_ID):** This field is a vendor-defined ID number that indicates the nature and format of the VSEC structure. Software must qualify the Vendor ID before interpreting this field.

16.678 SW LTR POINTER (SWLTRPTR)—Offset 98h

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 21
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**SW LTR Update MMIO Offset Location (SW_LTR_UPDT_MMIO_OFFSET):** This register contains the location pointer to the SW LTR register in MMIO space, as an offset from the specified BAR. The value of this register is a don't care, if the Valid bit is not set.
3:1	0h RO	**Base Address Register Number (BARNUM):** Contains the 0's based AR Number of the BAR which contains the location of the SW LTR MMIO register. When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number. Bar 0 is the 1st BAR. The value of this register is a don't care, if the Valid bit is not set.
0	0h RO	**Valid (VALID):** Set to '1' to indicate that the function has implemented a SW LTR register as specified in the DevIdle that can be located using the SWLTRLOC register and BARNUM. Set to '0' to indicate that the function has not implemented a SW LTR register that is compliant to the DevIdle definition. This could be because the function has not implemented SW LTR at all, or has a device specific version of SW LTR.

16.679 Device Idle Pointer Register (DEVIDLEPTR)—Offset 9Ch

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 21
Function: 0

Default: 80AC1h

Bit Range	Default & Access	Field Name (ID): Description
31:4	80ACh RW/L	**DevIdle MMIO Offset Location (DEVIDLELOC):** This register contains the location pointer to the DevIdle register in MMIO space, as an offset from the specified BAR. The value of this register is a don't care, if the Valid bit is not set.
3:1	0h RO	**Base Address Register Number (BARNUM):** Contains the 0's based BAR Number of the BAR which contains the location of the DevIdle MMIO register. When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number. Bar 0 is the 1st BAR. The value of this register is a don't care, if the Valid bit is not set.
0	1h RW/L	**Valid (VALID):** Set to '1' to indicate that the function has implemented a DevIdle register as specified in the DevIdle that can be located using the DEVIDLELOC register and BARNUM. Set to '0' to indicate that the function has not implemented a DevIdle register that is compliant to the DevIdle definition. This could be because the function has not implemented DevIdle at all, or has a device specific version of DevIdle.

16.680 Device Idle Power ON Latency (DEVIDLEPOL)—Offset A0h

Access Method

Type: CFG Register (Size: 16 bits)
Device: 21
Function: 0

Default: 800h

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	**Rsvd1 (Rsvd1)**
12:10	2h RW/L	**Power On Latency Scale (POLS):** Latency Scale multiplier: 010: 1us 011: 32us All other settings are reserved. The value of this register is multiplied with the Power On Latency Value (POLV) to provide the DevIdle power on exit latency multiplier scale from 1us to 32ms. This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value. This register is undefined if the DevIdle.Valid bit is '0'.
9:0	0h RW/L	**Power On Latency Value (POLV):** 10-bit value that is multiplied by the Power On Latency Scale. A value of 0 indicates a power on latency of less than 1us. This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value. This register is undefined if the DevIdle.Valid bit is '0'.

16.681 High Speed Configuration 2 (HSCFG2)—Offset A4h

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 21
Function: 0

Default: 2000h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Rsvd1 (Rsvd1)**
18	0h RW	**PORT1_HOST_MODE_OVERRIDE (PORT1_HOST_MODE_OVERRIDE):** When set, this bit causes the Host_Device mux on port 1 to be forced into the Host mode.
17:16	0h RW	**eUSB2SEL (eUSB2SEL):** The two bits are associate with USB2 ports 1 - bit 16 and 2 - bit 2 0: Port is mapped to USB2 1: Port is mapped to eUSB2
15	0h RW	**HS ASYNC Active IN Mask (HSAAIM):** Determines if the Async Active will mask/ignore IN EP s. 0 HS ASYNC Active will include IN EP s. 1 HS ASYNC Active will mask/ignore IN EP s.
14	0h RW	**HS OUT ASYNC Active Polling EP Mask (HSOAAPEPM):** Determines if the Async Active for OUT HS/FS/LS masks/ignores EP s that are polling/PINGing (HS) due to NAK. 0 HS OUT ASYNC Active will include EP s that are polling. 1 HS OUT ASYNC Active will mask/ignore EP s that are polling.
13	1h RW	**HS IN ASYNC Active Polling EP Mask (HSIAAPEPM):** Determines if the Async Active for IN HS/FS/LS masks/ignores EP s that are polling due to NAK. 0 HS IN ASYNC Active will include EP s that are polling. 1 HS IN ASYNC Active will mask/ignore EP s that are polling.
12:11	0h RW	**HS INTR IN Periodic Active Policy Control (HSIIPAPC):** Controls how the HS INTR IN periodic active is used to generate the global periodic active. This will determine how the smallest service interval among active EP s and number of active EP s are used. 0 HS INTR IN periodic active will be used to generate periodic active if Service Interval Threshold OR Numb of EP Threshold values meet the requirement. 1 HS INTR IN periodic active will be used to generate periodic active if Service Interval Threshold AND Numb of EP Threshold values meet the requirement. 2 Always allow HS INTR EP s to be used in the generation of the global Periodic Active indication. 3 Never allow HS INTR EP s to be used in the generation of the global Periodic Active indicaiton
10:4	0h RW	**HS INTR IN Periodic Active Num of EP Threshold (HSIIPANEPT):** Defines the threshold used to determine if Periodic Acive may include HS/FS/LS INTR IN EP active indication. If there are more than NumEPThreshold active HS/FS/LS INTR EP s then they may be included as part of the periodic active generation.

Bit Range	Default & Access	Field Name (ID): Description
3:0	0h RW	**HS INTR IN Periodic Active Service Interval Threshold (HSIIPASIT):** Defines the Service Interval threshold used to determine if Periodic Acive will include HS/FS/LS INTR IN EP active indication. If there are any active HS/FS/LS INTR EP s with a service interval less than or equal to this threshold then they may be included as part of the periodic active generation.

16.682 XHCI USB2 Overcurrent Pin Mapping 1 (U2OCM1)—Offset B0h

The RW/L property of this register is controlled by OCCFDONE bit.

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Rsvd (Rsvd):** Reserved
8:0	0h RW/L	**OC Mapping (OCM)**

16.683 XHCI USB2 Overcurrent Pin Mapping 2 (U2OCM2)—Offset B4h

Reserved

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Rsvd (Rsvd):** Reserved
8:0	0h RW/L	**OC Mapping (OCM)**

16.684 XHCI USB3 Overcurrent Pin Mapping 1 (U3OCM1)—Offset D0h

The RW/L property of this register is controlled by OCCFDONE bit.

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Rsvd (Rsvd)**
6:0	0h RW/L	**OC Mapping (OCM)**

16.685 XHCI USB3 Overcurrent Pin Mapping 2 (U3OCM2)—Offset D4h

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Rsvd (Rsvd)**
6:0	0h RW/L	**OC Mapping (OCM)**

16.686 XHCC3 (XHCC3)—Offset FCh

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 0

Default: 2h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (Rsvd1):** Reserved
3	0h RW	**Error Handling : Disable Command Parity Check (DISABLE_COMMAND_PARITY_CHECK):** When set to 1, XHCI Host Controller disables checking of Command Parity on command received as a target on its IOSF Primary interface

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Error Handling : Disable Data Parity Check (DISABLE_DATA_PARITY_CHECK):** When set to 1, XHCI Host Controller disables checking of Data Parity on data received as a target on its IOSF Primary interface
1	1h RW	**Error Handling : Enable ECC Error Response (ENABLE_ECC_ERROR_RESPONSE):** When set to 1, XHCI Host Controller will check for ECC on RFs (that support ECC) and halt operation when uncorrectable ECC is detected
0	0h RW/L	**Function Disable (FXN_DISABLE):** When set will disable the xHC from being operational.

16.687 Capability Registers Length (CAPLENGTH)—Offset 0h

This register is modified and maintained by BIOS

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 80h

Bit Range	Default & Access	Field Name (ID): Description
7:0	80h RW/L	**Capability Registers Length (CAPLENGTH)**

16.688 Host Controller Interface Version Number (HCIVERSION)—Offset 2h

This register is modified and maintained by BIOS

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
15:0	100h RW/L	**Host Controller Interface Version Number (HCIVERSION)**

16.689 Structural Parameters 1 (HCSPARAMS1)—Offset 4h

This register is modified and maintained by BIOS

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 10000820h

Bit Range	Default & Access	Field Name (ID): Description
31:24	10h RW/L	**Number of Ports (MaxPorts)**
23:19	0h RW/L	**Rsvd1 (Rsvd1)**
18:8	8h RW/L	**Number of Interrupters (MaxIntrs)**
7:0	20h RW/L	**Number of Device Slots (MaxSlots)**

16.690 Structural Parameters 2 (HCSPARAMS2)—Offset 8h

This register is modified and maintained by BIOS

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 94000054h

Bit Range	Default & Access	Field Name (ID): Description
31:27	12h RW/L	**Max Scratchpad Buffers LO (MaxScratchpadBufs)**
26	1h RW/L	**Scratchpad Restore (SPR)**
25:21	0h RW/L	**Max Scratchpad Buffers HI (MaxScratchpadBufs_HI)**
20:8	0h RW/L	**Rsvd1 (Rsvd1)**
7:4	5h RW/L	**Event Ring Segment Table Max (ERSTMax)**
3:0	4h RW/L	**Isochronous Scheduling Threshold (IST)**

16.691 Structural Parameters 3 (HCSPARAMS3)—Offset Ch

This register is modified and maintained by BIOS

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 40001h

Bit Range	Default & Access	Field Name (ID): Description
31:16	4h RW/L	**U2 Device Exit Latency (U2DEL)**
15:8	0h RW/L	**Rsvd1 (Rsvd1)**
7:0	1h RW/L	**U1 Device Exit Latency (U1DEL)**

16.692 Capability Parameters (HCCPARAMS)—Offset 10h

This register is modified and maintained by BIOS

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 200077C1h

Bit Range	Default & Access	Field Name (ID): Description
31:16	2000h RW/L	**xHCI Extended Capabilities Pointer (xECP)**
15:12	7h RW/L	**Maximum Primary Stream Array Size (MaxPSASize)**
11	0h RW/L	**Contiguous Frame ID Capability (CFC)**
10	1h RW/L	**Stopped EDLTA Capabilty (SEC)**
9	1h RW/L	**Stopped - Short Packet Capability (SPC)**
8	1h RW/L	**Parst All Event Data (PAE)**
7	1h RW/L	**No Secondary SID Support (NSS)**
6	1h RW/L	**Latency Tolerance Messaging Capability (LTC)**
5	0h RW/L	**Light HC Reset Capability (LHRC)**
4	0h RW/L	**Port Indicators (PIND)**
3	0h RW/L	**Port Power Control (PPC)**
2	0h RW/L	**Context Size (CSZ)**
1	0h RW/L	**BW Negotiation Capability (BNC)**
0	1h RW/L	**64-bit Addressing Capability (AC64)**

16.693 Doorbell Offset (DBOFF)—Offset 14h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:2	C00h RO	**Doorbell Array Offset (DBAO)**
1:0	0h RO	**Rsvd1 (Rsvd1)**

16.694 Runtime Register Space Offset (RTSOFF)—Offset 18h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2000h

Bit Range	Default & Access	Field Name (ID): Description
31:5	100h RO	**Runtime Register Space Offset (RTRSO)**
4:0	0h RO	**Rsvd1 (Rsvd1)**

16.695 USB Command (USBCMD)—Offset 80h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Rsvd2 (Rsvd2)**
11	0h RW	**Enable U3 MFINDEX Stop (EU3S)**
10	0h RW	**Enable Wrap Event (EWE)**
9	0h RW	**Controller Restore State (CRS)**

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW	**Controller Save State (CSS)**
7	0h RW	**Light Host Controller Reset (LHCRST)**
6:4	0h RO	**Rsvd1 (Rsvd1)**
3	0h RW	**Host System Error Enable (HSEE)**
2	0h RW	**Interrupter Enable (INTE)**
1	0h RW	**Host Controller Reset (HCRST)**
0	0h RW	**Run/Stop (RS)**

16.696 USB Status (USBSTS)—Offset 84h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
31:13	0h RO	**Rsvd3 (Rsvd3)**
12	0h RO	**Host Controller Error (HCE):** This bit is not preset in HC, this is deviation from XHCI 1.0 spec.
11	0h RO	**Controller Not Ready (CNR):** This is deviation from XHCI 1.0 spec.
10	0h RW/C	**Save/Restore Error (SRE)**
9	0h RO	**Restore State Status (RSS)**
8	0h RO	**Save State Status (SSS)**
7:5	0h RO	**Rsvd2 (Rsvd2)**
4	0h RW/C	**Port Change Detect (PCD)**
3	0h RW/C	**Event Interrupt (EINT)**
2	0h RW/C	**Host System Error (HSE)**
1	0h RO	**Rsvd1 (Rsvd1)**

Bit Range	Default & Access	Field Name (ID): Description
0	1h RO	**HCHalted (HCH)**

16.697 Page Size (PAGESIZE)—Offset 88h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	1h RO	**Page Size (PAGESIZE)**

16.698 Device Notification Control (DNCTRL)—Offset 94h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RW	**Notification Enable (N0_N15)**

16.699 Command Ring Low (CRCR_LO)—Offset 98h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Command Ring Pointer (CRP)**

Bit Range	Default & Access	Field Name (ID): Description
5:4	0h RO	**Rsvd1 (Rsvd1)**
3	0h RO	**Command Ring Running (CRR)**
2	0h RW/1S	**Command Abort (CA)**
1	0h RW/1S	**Command Stop (CS)**
0	0h RW	**Ring Cycle State (RCS)**

16.700 Command Ring High (CRCR_HI)—Offset 9Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Command Ring Pointer (CRP)**

16.701 Device Context Base Address Array Pointer Low (DCBAAP_LO)—Offset B0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Device Context Base Address Array Pointer (DCBAAP)**
5:0	0h RO	**Rsvd1 (Rsvd1)**

16.702 Device Context Base Address Array Pointer High (DCBAAP_HI)—Offset B4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Device Context Base Address Array Pointer (DCBAAP)**

16.703 Configure (CONFIG)—Offset B8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**Max Device Slots Enabled (MaxSlotsEn)**

16.704 Port Status and Control USB2 (PORTSC1)—Offset 480h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.705 Port Power Management Status and Control USB2 (PORTPMSC1)—Offset 484h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Port Test Control (PTC):** Note: This register is sticky.
27:17	0h RO	**Rsvd1 (Rsvd1)**
16	0h RW	**Hardware LPM Enable (HLE)**
15:8	0h RW	**Device Address (DA):** Note: This register is sticky.
7:4	0h RW	**Host Initiated Resume Duration (HIRD):** Note: This register is sticky.
3	0h RW	**Remote Wake Enable (RWE):** Note: This register is sticky.
2:0	0h RW	**L1 Status (L1S):** Note: This register is sticky.

16.706 Port X Hardware LPM Control Register (PORTHLPMC1)—Offset 48Ch

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters neccessary for xHC to automatically generate an LPM Token to the downstream device.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**RESERVED (RSVD)**
13:10	0h RW	**Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1. The HIRDD value is is encoded as follows: 0h: 50 us (default) 1h: 125 us 2h: 200 us ... Fh: 1.175ms The value of 0h is interpreted as 50 us. Each incrementing value adds 75 us to the previous value.
9:2	0h RW	**L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer. This field shall be set to 00h by assertion of PR to '1'. Following are permissible values: 00h: 128 us (default) 01h: 256 us. ... FFh: 65,280us Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used. Following are permissible values: 0: Initiate L1 using HIRD only time out (default) 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD 2,3: Reserved Note: This register is sticky.

16.707 Port Status and Control USB2 (PORTSC2)—Offset 490h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.708 Port Power Management Status and Control USB2 (PORTPMSC2)—Offset 494h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Port Test Control (PTC):** Note: This register is sticky.
27:17	0h RO	**Rsvd1 (Rsvd1)**
16	0h RW	**Hardware LPM Enable (HLE)**
15:8	0h RW	**Device Address (DA):** Note: This register is sticky.
7:4	0h RW	**Host Initiated Resume Duration (HIRD):** Note: This register is sticky.
3	0h RW	**Remote Wake Enable (RWE):** Note: This register is sticky.
2:0	0h RW	**L1 Status (L1S):** Note: This register is sticky.

16.709 Port X Hardware LPM Control Register (PORTHLPMC2)—Offset 49Ch

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters neccessary for xHC to automatically generate an LPM Token to the downstream device.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**RESERVED (RSVD)**
13:10	0h RW	**Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1. The HIRDD value is is encoded as follows: 0h: 50 us (default) 1h: 125 us 2h: 200 us ... Fh: 1.175ms The value of 0h is interpreted as 50 us. Each incrementing value adds 75 us to the previous value.
9:2	0h RW	**L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer. This field shall be set to 00h by assertion of PR to '1'. Following are permissible values: 00h: 128 us (default) 01h: 256 us. ... FFh: 65,280us Note: This register is sticky.
1:0	0h RW	**Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used. Following are permissible values: 0: Initiate L1 using HIRD only time out (default) 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD 2,3: Reserved Note: This register is sticky.

16.710 Port Status and Control USB2 (PORTSC3)—Offset 4A0h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**

Bit Range	Default & Access	Field Name (ID): Description
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.711 Port Power Management Status and Control USB2 (PORTPMSC3)—Offset 4A4h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Port Test Control (PTC):** Note: This register is sticky.
27:17	0h RO	**Rsvd1 (Rsvd1)**
16	0h RW	**Hardware LPM Enable (HLE)**
15:8	0h RW	**Device Address (DA):** Note: This register is sticky.
7:4	0h RW	**Host Initiated Resume Duration (HIRD):** Note: This register is sticky.
3	0h RW	**Remote Wake Enable (RWE):** Note: This register is sticky.
2:0	0h RW	**L1 Status (L1S):** Note: This register is sticky.

16.712 Port X Hardware LPM Control Register (PORTHLPMC3)—Offset 4ACh

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters neccessary for xHC to automatically generate an LPM Token to the downstream device.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**RESERVED (RSVD)**

Bit Range	Default & Access	Field Name (ID): Description
13:10	0h RW	**Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1. The HIRDD value is is encoded as follows: 0h: 50 us (default) 1h: 125 us 2h: 200 us ... Fh: 1.175ms The value of 0h is interpreted as 50 us. Each incrementing value adds 75 us to the previous value.
9:2	0h RW	**L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer. This field shall be set to 00h by assertion of PR to '1'. Following are permissible values: 00h: 128 us (default) 01h: 256 us. ... FFh: 65,280us Note: This register is sticky.
1:0	0h RW	**Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used. Following are permissible values: 0: Initiate L1 using HIRD only time out (default) 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD 2,3: Reserved Note: This register is sticky.

16.713 Port Status and Control USB2 (PORTSC4)—Offset 4B0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.714 Port Power Management Status and Control USB2 (PORTPMSC4)—Offset 4B4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Port Test Control (PTC):** Note: This register is sticky.
27:17	0h RO	**Rsvd1 (Rsvd1)**
16	0h RW	**Hardware LPM Enable (HLE)**

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RW	**Device Address (DA):** Note: This register is sticky.
7:4	0h RW	**Host Initiated Resume Duration (HIRD):** Note: This register is sticky.
3	0h RW	**Remote Wake Enable (RWE):** Note: This register is sticky.
2:0	0h RW	**L1 Status (L1S):** Note: This register is sticky.

16.715 Port X Hardware LPM Control Register (PORTHLPMC4)—Offset 4BCh

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters neccessary for xHC to automatically generate an LPM Token to the downstream device.

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**RESERVED (RSVD)**
13:10	0h RW	**Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1. The HIRDD value is is encoded as follows: 0h: 50 us (default) 1h: 125 us 2h: 200 us ... Fh: 1.175ms The value of 0h is interpreted as 50 us. Each incrementing value adds 75 us to the previous value.
9:2	0h RW	**L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer. This field shall be set to 00h by assertion of PR to '1'. Following are permissible values: 00h: 128 us (default) 01h: 256 us. ... FFh: 65,280us Note: This register is sticky.
1:0	0h RW	**Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used. Following are permissible values: 0: Initiate L1 using HIRD only time out (default) 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD 2,3: Reserved Note: This register is sticky.

16.716 Port Status and Control USB2 (PORTSC5)—Offset 4C0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.717 Port Power Management Status and Control USB2 (PORTPMSC5)—Offset 4C4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Port Test Control (PTC):** Note: This register is sticky.
27:17	0h RO	**Rsvd1 (Rsvd1)**
16	0h RW	**Hardware LPM Enable (HLE)**
15:8	0h RW	**Device Address (DA):** Note: This register is sticky.
7:4	0h RW	**Host Initiated Resume Duration (HIRD):** Note: This register is sticky.
3	0h RW	**Remote Wake Enable (RWE):** Note: This register is sticky.
2:0	0h RW	**L1 Status (L1S):** Note: This register is sticky.

16.718 Port X Hardware LPM Control Register (PORTHLPMC5)—Offset 4CCh

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters neccessary for xHC to automatically generate an LPM Token to the downstream device.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**RESERVED (RSVD)**
13:10	0h RW	**Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1. The HIRDD value is is encoded as follows: 0h: 50 us (default) 1h: 125 us 2h: 200 us ... Fh: 1.175ms The value of 0h is interpreted as 50 us. Each incrementing value adds 75 us to the previous value.
9:2	0h RW	**L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer. This field shall be set to 00h by assertion of PR to '1'. Following are permissible values: 00h: 128 us (default) 01h: 256 us. ... FFh: 65,280us Note: This register is sticky.
1:0	0h RW	**Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used. Following are permissible values: 0: Initiate L1 using HIRD only time out (default) 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD 2,3: Reserved Note: This register is sticky.

16.719 Port Status and Control USB2 (PORTSC6)—Offset 4D0h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.720 Port Power Management Status and Control USB2 (PORTPMSC6)—Offset 4D4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Port Test Control (PTC):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
27:17	0h RO	**Rsvd1 (Rsvd1)**
16	0h RW	**Hardware LPM Enable (HLE)**
15:8	0h RW	**Device Address (DA):** Note: This register is sticky.
7:4	0h RW	**Host Initiated Resume Duration (HIRD):** Note: This register is sticky.
3	0h RW	**Remote Wake Enable (RWE):** Note: This register is sticky.
2:0	0h RW	**L1 Status (L1S):** Note: This register is sticky.

16.721 Port X Hardware LPM Control Register (PORTHLPMC6)—Offset 4DCh

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters neccessary for xHC to automatically generate an LPM Token to the downstream device.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**RESERVED (RSVD)**
13:10	0h RW	**Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1. The HIRDD value is is encoded as follows: 0h: 50 us (default) 1h: 125 us 2h: 200 us ... Fh: 1.175ms The value of 0h is interpreted as 50 us. Each incrementing value adds 75 us to the previous value.
9:2	0h RW	**L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer. This field shall be set to 00h by assertion of PR to '1'. Following are permissible values: 00h: 128 us (default) 01h: 256 us. ... FFh: 65,280us Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used. Following are permissible values: 0: Initiate L1 using HIRD only time out (default) 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD 2,3: Reserved Note: This register is sticky.

16.722 Port Status and Control USB2 (PORTSC7)—Offset 4E0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.723 Port Power Management Status and Control USB2 (PORTPMSC7)—Offset 4E4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Port Test Control (PTC):** Note: This register is sticky.
27:17	0h RO	**Rsvd1 (Rsvd1)**
16	0h RW	**Hardware LPM Enable (HLE)**
15:8	0h RW	**Device Address (DA):** Note: This register is sticky.
7:4	0h RW	**Host Initiated Resume Duration (HIRD):** Note: This register is sticky.
3	0h RW	**Remote Wake Enable (RWE):** Note: This register is sticky.
2:0	0h RW	**L1 Status (L1S):** Note: This register is sticky.

16.724 Port X Hardware LPM Control Register (PORTHLPMC7)—Offset 4ECh

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters neccessary for xHC to automatically generate an LPM Token to the downstream device.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**RESERVED (RSVD)**
13:10	0h RW	**Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1. The HIRDD value is is encoded as follows: 0h: 50 us (default) 1h: 125 us 2h: 200 us ... Fh: 1.175ms The value of 0h is interpreted as 50 us. Each incrementing value adds 75 us to the previous value.
9:2	0h RW	**L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer. This field shall be set to 00h by assertion of PR to '1'. Following are permissible values: 00h: 128 us (default) 01h: 256 us. ... FFh: 65,280us Note: This register is sticky.
1:0	0h RW	**Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used. Following are permissible values: 0: Initiate L1 using HIRD only time out (default) 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD 2,3: Reserved Note: This register is sticky.

16.725 Port Status and Control USB2 (PORTSC8)—Offset 4F0h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**

Bit Range	Default & Access	Field Name (ID): Description
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.726 Port Power Management Status and Control USB2 (PORTPMSC8)—Offset 4F4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Port Test Control (PTC):** Note: This register is sticky.
27:17	0h RO	**Rsvd1 (Rsvd1)**
16	0h RW	**Hardware LPM Enable (HLE)**
15:8	0h RW	**Device Address (DA):** Note: This register is sticky.
7:4	0h RW	**Host Initiated Resume Duration (HIRD):** Note: This register is sticky.
3	0h RW	**Remote Wake Enable (RWE):** Note: This register is sticky.
2:0	0h RW	**L1 Status (L1S):** Note: This register is sticky.

16.727 Port X Hardware LPM Control Register (PORTHLPMC8)—Offset 4FCh

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters neccessary for xHC to automatically generate an LPM Token to the downstream device.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**RESERVED (RSVD)**

Bit Range	Default & Access	Field Name (ID): Description
13:10	0h RW	**Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1. The HIRDD value is is encoded as follows: 0h: 50 us (default) 1h: 125 us 2h: 200 us ... Fh: 1.175ms The value of 0h is interpreted as 50 us. Each incrementing value adds 75 us to the previous value.
9:2	0h RW	**L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer. This field shall be set to 00h by assertion of PR to '1'. Following are permissible values: 00h: 128 us (default) 01h: 256 us. ... FFh: 65,280us Note: This register is sticky.
1:0	0h RW	**Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used. Following are permissible values: 0: Initiate L1 using HIRD only time out (default) 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD 2,3: Reserved Note: This register is sticky.

16.728 Port Status and Control USB2 (PORTSC9)—Offset 500h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.729 Port Power Management Status and Control USB2 (PORTPMSC9)—Offset 504h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Port Test Control (PTC):** Note: This register is sticky.
27:17	0h RO	**Rsvd1 (Rsvd1)**
16	0h RW	**Hardware LPM Enable (HLE)**

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RW	**Device Address (DA):** Note: This register is sticky.
7:4	0h RW	**Host Initiated Resume Duration (HIRD):** Note: This register is sticky.
3	0h RW	**Remote Wake Enable (RWE):** Note: This register is sticky.
2:0	0h RW	**L1 Status (L1S):** Note: This register is sticky.

16.730 Port X Hardware LPM Control Register (PORTHLPMC9)—Offset 50Ch

There are 9 PORTHLPMC registers at offsets 48Ch, 49Ch, 4ACh, 4BCh, 4CCh, 4DCh, 4ECh, 4FCh, 50Ch This register is reset only by platform hardware during cold reset or in response to a Host Controller Reset (HCRST). The definition for the fields depend on the protocol supported. For USB3 this register is reserved and shall be treated by software as RsvdP. For USB2 the definition is given below. Fields contain parameters neccessary for xHC to automatically generate an LPM Token to the downstream device.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**RESERVED (RSVD)**
13:10	0h RW	**Host Initiated Resume Duration-Deep (HIRDD):** System software sets this field to indicate to the recipient device how long the xHC will drive resume if an exit from L1. The HIRDD value is is encoded as follows: 0h: 50 us (default) 1h: 125 us 2h: 200 us ... Fh: 1.175ms The value of 0h is interpreted as 50 us. Each incrementing value adds 75 us to the previous value.
9:2	0h RW	**L1 Timeout (L1_TO):** Timeout value for L1 inactivity timer. This field shall be set to 00h by assertion of PR to '1'. Following are permissible values: 00h: 128 us (default) 01h: 256 us. ... FFh: 65,280us Note: This register is sticky.
1:0	0h RW	**Host Initiated Resume Duration Mode (HIRDM):** Indicates which HIRD value should be used. Following are permissible values: 0: Initiate L1 using HIRD only time out (default) 1: Initiate HIRDDon timeout. If rejected by device, initiate L1 using HIRD 2,3: Reserved Note: This register is sticky.

16.731 Port Status and Control USB3 (PORTSC10)—Offset 510h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.732 Port Power Management Status and Control USB3 (PORTPMSC10)—Offset 514h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RSVD)**
16	0h RW	**Force Link PM Accept (FLA)**
15:8	0h RW/S	**U2 Timeout (U2T)**
7:0	0h RW/S	**U1 Timeout (U1T)**

16.733 USB3 Port Link Info (PORTLI10)—Offset 518h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RO	**Link Error Count (LEC)**

16.734 Port Status and Control USB3 (PORTSC11)—Offset 520h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.735 Port Power Management Status and Control USB3 (PORTPMSC11)—Offset 524h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RSVD)**
16	0h RW	**Force Link PM Accept (FLA)**
15:8	0h RW/S	**U2 Timeout (U2T)**
7:0	0h RW/S	**U1 Timeout (U1T)**

16.736 USB3 Port Link Info (PORTLI11)—Offset 528h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RO	**Link Error Count (LEC)**

16.737 Port Status and Control USB3 (PORTSC12)—Offset 530h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.738 Port Power Management Status and Control USB3 (PORTPMSC12)—Offset 534h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RSVD)**
16	0h RW	**Force Link PM Accept (FLA)**
15:8	0h RW/S	**U2 Timeout (U2T)**
7:0	0h RW/S	**U1 Timeout (U1T)**

16.739 USB3 Port Link Info (PORTLI12)—Offset 538h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RO	**Link Error Count (LEC)**

16.740 Port Status and Control USB3 (PORTSC13)—Offset 540h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.741 Port Power Management Status and Control USB3 (PORTPMSC13)—Offset 544h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RSVD)**
16	0h RW	**Force Link PM Accept (FLA)**
15:8	0h RW/S	**U2 Timeout (U2T)**
7:0	0h RW/S	**U1 Timeout (U1T)**

16.742 USB3 Port Link Info (PORTLI13)—Offset 548h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RO	**Link Error Count (LEC)**

16.743 Port Status and Control USB3 (PORTSC14)—Offset 550h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.744 Port Power Management Status and Control USB3 (PORTPMSC14)—Offset 554h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RSVD)**
16	0h RW	**Force Link PM Accept (FLA)**
15:8	0h RW/S	**U2 Timeout (U2T)**
7:0	0h RW/S	**U1 Timeout (U1T)**

16.745 USB3 Port Link Info (PORTLI14)—Offset 558h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RO	**Link Error Count (LEC)**

16.746 Port Status and Control USB3 (PORTSC15)—Offset 560h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.747 Port Power Management Status and Control USB3 (PORTPMSC15)—Offset 564h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RSVD)**
16	0h RW	**Force Link PM Accept (FLA)**
15:8	0h RW/S	**U2 Timeout (U2T)**
7:0	0h RW/S	**U1 Timeout (U1T)**

16.748 USB3 Port Link Info (PORTLI15)—Offset 568h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RO	**Link Error Count (LEC)**

16.749 Port Status and Control USB3 (PORTSC16)—Offset 570h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2A0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S	**Warm Port Reset (WPR)**
30	0h RW/L	**Device Removable (DR)**
29:28	0h RO	**Rsvd2 (Rsvd2)**
27	0h RW	**Wake on Over-current Enable (WOE):** Note: This register is sticky.
26	0h RW	**Wake on Disconnect Enable (WDE):** Note: This register is sticky.
25	0h RW	**Wake on Connect Enable (WCE):** Note: This register is sticky.
24	0h RO	**Cold Attach Status (CAS)**
23	0h RW/C	**Port Config Error Change (CEC):** Note: This register is sticky.
22	0h RW/C	**Port Link State Change (PLC):** Note: This register is sticky.
21	0h RW/C	**Port Reset Change (PRC):** Note: This register is sticky.
20	0h RW/C	**Over-current Change (OCC):** Note: This register is sticky.
19	0h RW/C	**Warm Port Reset Change (WRC):** Note: This register is sticky.
18	0h RW/C	**Port Enabled Disabled Change (PEC):** Note: This register is sticky.
17	0h RW/C	**Connect Status Change (CSC):** Note: This register is sticky.
16	0h RW	**Port Link State Write Strobe (LWS)**
15:14	0h RW	**Port Indicator Control (PIC):** Note: This register is sticky.
13:10	0h RW	**Port Speed (PortSpeed):** Note: This register is sticky.
9	1h RW	**Port Power (PP):** Note: This register is sticky.
8:5	5h RW	**Port Link State (PLS):** Note: This register is sticky.
4	0h RW/1S	**Port Reset (PR)**
3	0h RW	**Over-current Active (OCA):** Note: This register is sticky.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW/C	**Port Enabled Disabled (PED):** Note: This register is sticky.
0	0h RW	**Current Connect Status (CCS):** Note: This register is sticky.

16.750 Port Power Management Status and Control USB3 (PORTPMSC16)—Offset 574h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RSVD)**
16	0h RW	**Force Link PM Accept (FLA)**
15:8	0h RW/S	**U2 Timeout (U2T)**
7:0	0h RW/S	**U1 Timeout (U1T)**

16.751 USB3 Port Link Info (PORTLI16)—Offset 578h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RO	**Link Error Count (LEC)**

16.752 Microframe Index (RTMFINDEX)—Offset 2000h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**Rsvd1 (Rsvd1)**
13:0	0h RO	**Microframe Index (IMAN0)**

16.753 Interrupter 1 Management (IMAN0)—Offset 2020h

There are 8 IMAN registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW	**Interrupt Enable (IE)**
0	0h RW/C	**Interrupt Pending (IP)**

16.754 Interrupter 1 Moderation (IMOD0)—Offset 2024h

There are 8 IMOD registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: FA0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Interrupt Moderation Counter (IMODC)**
15:0	FA0h RW	**Interrupt Moderation Interval (IMODI)**

16.755 Event Ring Segment Table Size 1 (ERSTSZ0)—Offset 2028h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RW	**Event Ring Segment Table Size (ERSTS)**

16.756 Event Ring Segment Table Base Address Low 1 (ERSTBA_LO0)—Offset 2030h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Event Ring Segment Table Base Address Register (ERSTBA)**
5:0	0h RO	**Rsvd1 (Rsvd1)**

16.757 Event Ring Segment Table Base Address High 1 (ERSTBA_HI0)—Offset 2034h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Segment Table Base Address (ERSTBA)**

16.758 Event Ring Dequeue Pointer Low 1 (ERDP_LO0)—Offset 2038h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RW	**Event Ring Dequeue Pointer (ERDP)**
3	0h RW/C	**Event Handler Busy (EHB)**
2:0	0h RW	**Dequeue ERST Segment Index (DESI)**

16.759 Event Ring Dequeue Pointer High 1 (ERDP_HI0)—Offset 203Ch

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Dequeue Pointer (ERDP)**

16.760 Interrupter 2 Management (IMAN1)—Offset 2040h

There are 8 IMAN registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW	**Interrupt Enable (IE)**
0	0h RW/C	**Interrupt Pending (IP)**

16.761 Interrupter 2 Moderation (IMOD1)—Offset 2044h

There are 8 IMOD registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: FA0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Interrupt Moderation Counter (IMODC)**
15:0	FA0h RW	**Interrupt Moderation Interval (IMODI)**

16.762 Event Ring Segment Table Size 2 (ERSTSZ1)—Offset 2048h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RW	**Event Ring Segment Table Size (ERSTS)**

16.763 Event Ring Segment Table Base Address Low 2 (ERSTBA_LO1)—Offset 2050h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Event Ring Segment Table Base Address Register (ERSTBA)**
5:0	0h RO	**Rsvd1 (Rsvd1)**

16.764 Event Ring Segment Table Base Address High 2 (ERSTBA_HI1)—Offset 2054h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Segment Table Base Address (ERSTBA)**

16.765 Event Ring Dequeue Pointer Low 2 (ERDP_LO1)—Offset 2058h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RW	**Event Ring Dequeue Pointer (ERDP)**
3	0h RW/C	**Event Handler Busy (EHB)**
2:0	0h RW	**Dequeue ERST Segment Index (DESI)**

16.766 Event Ring Dequeue Pointer High 2 (ERDP_HI1)—Offset 205Ch

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Dequeue Pointer (ERDP)**

16.767 Interrupter 3 Management (IMAN2)—Offset 2060h

There are 8 IMAN registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW	**Interrupt Enable (IE)**
0	0h RW/C	**Interrupt Pending (IP)**

16.768 Interrupter 3 Moderation (IMOD2)—Offset 2064h

There are 8 IMOD registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: FA0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Interrupt Moderation Counter (IMODC)**

Bit Range	Default & Access	Field Name (ID): Description
15:0	FA0h RW	**Interrupt Moderation Interval (IMODI)**

16.769 Event Ring Segment Table Size 3 (ERSTSZ2)—Offset 2068h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RW	**Event Ring Segment Table Size (ERSTS)**

16.770 Event Ring Segment Table Base Address Low 3 (ERSTBA_LO2)—Offset 2070h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Event Ring Segment Table Base Address Register (ERSTBA)**
5:0	0h RO	**Rsvd1 (Rsvd1)**

16.771 Event Ring Segment Table Base Address High 3 (ERSTBA_HI2)—Offset 2074h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Segment Table Base Address (ERSTBA)**

16.772 Event Ring Dequeue Pointer Low 3 (ERDP_LO2)—Offset 2078h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RW	**Event Ring Dequeue Pointer (ERDP)**
3	0h RW/C	**Event Handler Busy (EHB)**
2:0	0h RW	**Dequeue ERST Segment Index (DESI)**

16.773 Event Ring Dequeue Pointer High 3 (ERDP_HI2)—Offset 207Ch

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Dequeue Pointer (ERDP)**

16.774 Interrupter 4 Management (IMAN3)—Offset 2080h

There are 8 IMAN registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW	**Interrupt Enable (IE)**
0	0h RW/C	**Interrupt Pending (IP)**

16.775 Interrupter 4 Moderation (IMOD3)—Offset 2084h

There are 8 IMOD registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: FA0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Interrupt Moderation Counter (IMODC)**
15:0	FA0h RW	**Interrupt Moderation Interval (IMODI)**

16.776 Event Ring Segment Table Size 4 (ERSTSZ3)—Offset 2088h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RW	**Event Ring Segment Table Size (ERSTS)**

16.777 Event Ring Segment Table Base Address Low 4 (ERSTBA_LO3)—Offset 2090h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Event Ring Segment Table Base Address Register (ERSTBA)**
5:0	0h RO	**Rsvd1 (Rsvd1)**

16.778 Event Ring Segment Table Base Address High 4 (ERSTBA_HI3)—Offset 2094h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Segment Table Base Address (ERSTBA)**

16.779 Event Ring Dequeue Pointer Low 4 (ERDP_LO3)—Offset 2098h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RW	**Event Ring Dequeue Pointer (ERDP)**
3	0h RW/C	**Event Handler Busy (EHB)**
2:0	0h RW	**Dequeue ERST Segment Index (DESI)**

16.780 Event Ring Dequeue Pointer High 4 (ERDP_HI3)—Offset 209Ch

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Dequeue Pointer (ERDP)**

16.781 Interrupter 5 Management (IMAN4)—Offset 20A0h

There are 8 IMAN registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW	**Interrupt Enable (IE)**
0	0h RW/C	**Interrupt Pending (IP)**

16.782 Interrupter 5 Moderation (IMOD4)—Offset 20A4h

There are 8 IMOD registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: FA0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Interrupt Moderation Counter (IMODC)**

Bit Range	Default & Access	Field Name (ID): Description
15:0	FA0h RW	**Interrupt Moderation Interval (IMODI)**

16.783 Event Ring Segment Table Size 5 (ERSTSZ4)—Offset 20A8h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RW	**Event Ring Segment Table Size (ERSTS)**

16.784 Event Ring Segment Table Base Address Low 5 (ERSTBA_LO4)—Offset 20B0h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Event Ring Segment Table Base Address Register (ERSTBA)**
5:0	0h RO	**Rsvd1 (Rsvd1)**

16.785 Event Ring Segment Table Base Address High 5 (ERSTBA_HI4)—Offset 20B4h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Segment Table Base Address (ERSTBA)**

16.786 Event Ring Dequeue Pointer Low 5 (ERDP_LO4)—Offset 20B8h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RW	**Event Ring Dequeue Pointer (ERDP)**
3	0h RW/C	**Event Handler Busy (EHB)**
2:0	0h RW	**Dequeue ERST Segment Index (DESI)**

16.787 Event Ring Dequeue Pointer High 5 (ERDP_HI4)—Offset 20BCh

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Dequeue Pointer (ERDP)**

16.788 Interrupter 6 Management (IMAN5)—Offset 20C0h

There are 8 IMAN registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW	**Interrupt Enable (IE)**
0	0h RW/C	**Interrupt Pending (IP)**

16.789 Interrupter 6 Moderation (IMOD5)—Offset 20C4h

There are 8 IMOD registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: FA0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Interrupt Moderation Counter (IMODC)**
15:0	FA0h RW	**Interrupt Moderation Interval (IMODI)**

16.790 Event Ring Segment Table Size 6 (ERSTSZ5)—Offset 20C8h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RW	**Event Ring Segment Table Size (ERSTS)**

16.791 Event Ring Segment Table Base Address Low 6 (ERSTBA_LO5)—Offset 20D0h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Event Ring Segment Table Base Address Register (ERSTBA)**
5:0	0h RO	**Rsvd1 (Rsvd1)**

16.792 Event Ring Segment Table Base Address High 6 (ERSTBA_HI5)—Offset 20D4h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Segment Table Base Address (ERSTBA)**

16.793 Event Ring Dequeue Pointer Low 6 (ERDP_LO5)—Offset 20D8h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RW	**Event Ring Dequeue Pointer (ERDP)**
3	0h RW/C	**Event Handler Busy (EHB)**

Bit Range	Default & Access	Field Name (ID): Description
2:0	0h RW	**Dequeue ERST Segment Index (DESI)**

16.794 Event Ring Dequeue Pointer High 6 (ERDP_HI5)—Offset 20DCh

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Dequeue Pointer (ERDP)**

16.795 Interrupter 7 Management (IMAN6)—Offset 20E0h

There are 8 IMAN registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW	**Interrupt Enable (IE)**
0	0h RW/C	**Interrupt Pending (IP)**

16.796 Interrupter 7 Moderation (IMOD6)—Offset 20E4h

There are 8 IMOD registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: FA0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Interrupt Moderation Counter (IMODC)**
15:0	FA0h RW	**Interrupt Moderation Interval (IMODI)**

16.797 Event Ring Segment Table Size 7 (ERSTSZ6)—Offset 20E8h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RW	**Event Ring Segment Table Size (ERSTS)**

16.798 Event Ring Segment Table Base Address Low 7 (ERSTBA_LO6)—Offset 20F0h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Event Ring Segment Table Base Address Register (ERSTBA)**
5:0	0h RO	**Rsvd1 (Rsvd1)**

16.799 Event Ring Segment Table Base Address High 7 (ERSTBA_HI6)—Offset 20F4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Segment Table Base Address (ERSTBA)**

16.800 Event Ring Dequeue Pointer Low 7 (ERDP_LO6)—Offset 20F8h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RW	**Event Ring Dequeue Pointer (ERDP)**
3	0h RW/C	**Event Handler Busy (EHB)**
2:0	0h RW	**Dequeue ERST Segment Index (DESI)**

16.801 Event Ring Dequeue Pointer High 7 (ERDP_HI6)—Offset 20FCh

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Dequeue Pointer (ERDP)**

16.802 Interrupter 8 Management (IMAN7)—Offset 2100h

There are 8 IMAN registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Rsvd1 (Rsvd1)**
1	0h RW	**Interrupt Enable (IE)**
0	0h RW/C	**Interrupt Pending (IP)**

16.803 Interrupter 8 Moderation (IMOD7)—Offset 2104h

There are 8 IMOD registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: FA0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Interrupt Moderation Counter (IMODC)**
15:0	FA0h RW	**Interrupt Moderation Interval (IMODI)**

16.804 Event Ring Segment Table Size 8 (ERSTSZ7)—Offset 2108h

There are 8 ERSTSZ register. x = 1, 2, ..., 8

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd1 (Rsvd1)**
15:0	0h RW	**Event Ring Segment Table Size (ERSTS)**

16.805 Event Ring Segment Table Base Address Low 8 (ERSTBA_LO7)—Offset 2110h

There are 8 ERSTBA_LO registers x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**Event Ring Segment Table Base Address Register (ERSTBA)**
5:0	0h RO	**Rsvd1 (Rsvd1)**

16.806 Event Ring Segment Table Base Address High 8 (ERSTBA_HI7)—Offset 2114h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Segment Table Base Address (ERSTBA)**

16.807 Event Ring Dequeue Pointer Low 8 (ERDP_LO7)—Offset 2118h

There are 8 ERDP_LO registers. x = 1, 2, ...,8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RW	**Event Ring Dequeue Pointer (ERDP)**
3	0h RW/C	**Event Handler Busy (EHB)**

Bit Range	Default & Access	Field Name (ID): Description
2:0	0h RW	**Dequeue ERST Segment Index (DESI)**

16.808 Event Ring Dequeue Pointer High 8 (ERDP_HI7)—Offset 211Ch

There are 8 ERDP_HI registers. x = 1, 2, ..., 8

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Event Ring Dequeue Pointer (ERDP)**

16.809 Door Bell 1 (DB0)—Offset 3000h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.810 Door Bell 2 (DB1)—Offset 3004h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.811 Door Bell 3 (DB2)—Offset 3008h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.812 Door Bell 4 (DB3)—Offset 300Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.813 Door Bell 5 (DB4)—Offset 3010h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.814 Door Bell 6 (DB5)—Offset 3014h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.815 Door Bell 7 (DB6)—Offset 3018h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.816 Door Bell 8 (DB7)—Offset 301Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.817 Door Bell 9 (DB8)—Offset 3020h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.818 Door Bell 10 (DB9)—Offset 3024h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.819 Door Bell 11 (DB10)—Offset 3028h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.820 Door Bell 12 (DB11)—Offset 302Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.821 Door Bell 13 (DB12)—Offset 3030h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.822 Door Bell 14 (DB13)—Offset 3034h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.823 Door Bell 15 (DB14)—Offset 3038h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.824 Door Bell 16 (DB15)—Offset 303Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.825 Door Bell 17 (DB16)—Offset 3040h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.826 Door Bell 18 (DB17)—Offset 3044h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.827 Door Bell 19 (DB18)—Offset 3048h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.828 Door Bell 20 (DB19)—Offset 304Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.829 Door Bell 21 (DB20)—Offset 3050h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.830 Door Bell 22 (DB21)—Offset 3054h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.831 Door Bell 23 (DB22)—Offset 3058h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.832 Door Bell 24 (DB23)—Offset 305Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.833 Door Bell 25 (DB24)—Offset 3060h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.834 Door Bell 26 (DB25)—Offset 3064h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.835 Door Bell 27 (DB26)—Offset 3068h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.836 Door Bell 28 (DB27)—Offset 306Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.837 Door Bell 29 (DB28)—Offset 3070h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.838 Door Bell 30 (DB29)—Offset 3074h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.839 Door Bell 31 (DB30)—Offset 3078h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.840 Door Bell 32 (DB31)—Offset 307Ch

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.841 Door Bell 32 (DB32)—Offset 3080h

Door Bell registers are an array of 64 registers, with 0 to 32 being used by the XHC and the rest being reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**DB Stream ID (DSI)**
15:8	0h RO	**Rsvd1 (Rsvd1)**
7:0	0h RW	**DB Target (DT)**

16.842 XECP_SUPP_USB2_0 (XECP_SUPP_USB2_0)—Offset 8000h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 2000802h

Bit Range	Default & Access	Field Name (ID): Description
31:24	2h RO	**USB Major Revision: 2.0 (USB2_MAJ_REV)**
23:16	0h RO	**USB Minor Revision (USB_MIN_REV)**
15:8	8h RO	**Next Capability Pointer (NCP)**
7:0	2h RO	**Supported Protocol ID (SPID)**

16.843 XECP_SUPP_USB2_1 (XECP_SUPP_USB2_1)—Offset 8004h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 20425355h

Bit Range	Default & Access	Field Name (ID): Description
31:0	20425355h RO	**XECP_SUPP_USB2_1 (XECP_SUPP_USB2_1):** Namestring USB

16.844 XECP_SUPP_USB2_2 (XECP_SUPP_USB2_2)—Offset 8008h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 30190901h

Bit Range	Default & Access	Field Name (ID): Description
31:28	3h RO	**Protocol Speed ID Count (PROT_SPD_ID_CNT):** 3 USB 2.0 Speed (High, Full, Low)
27:21	0h RO	**Rsvd0 (Rsvd0)**
20	1h RW/L	**BESL LPM Capability (BLC):** Bit is set to 1 to indicate that the the ports described by this xHCI Supported Protocol Capability will apply BESL timing to BESL and BESLD fields of the PORTPMSC and PORTHLPMCC registers.
19	1h RW/L	**Protocol Defined - Hardware LMP Capability (HLC)**
18	0h RO	**Protocol Defined - Integrated Hub Implementation (IHI)**

Bit Range	Default & Access	Field Name (ID): Description
17	0h RO	**Protocol Defined - High SPeed Only (HSO)**
16	1h RO	**Reserved (RSVD)**
15:8	9h RO	**Compatible Port Count (CPC)**
7:0	1h RO	**Compatible Port Offset (CPO)**

16.845 XECP_SUPP_USB2_3 (Full Speed) (XECP_SUPP_USB2_3)—Offset 8010h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: C0021h

Bit Range	Default & Access	Field Name (ID): Description
31:16	Ch RO	**Protocol Speed ID Mantissa (PSIM)**
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	0h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	2h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	1h RO	**Protocol Speed ID Value (PSIV)**

16.846 XECP_SUPP_USB2_4 (Low Speed) (XECP_SUPP_USB2_4)—Offset 8014h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 5DC0012h

Bit Range	Default & Access	Field Name (ID): Description
31:16	5DCh RO	**Protocol Speed ID Mantissa (PSIM)**

Bit Range	Default & Access	Field Name (ID): Description
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	0h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	1h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	2h RO	**Protocol Speed ID Value (PSIV)**

16.847 XECP_SUPP_USB2_5 (High Speed) (XECP_SUPP_USB2_5)—Offset 8018h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 1E00023h

Bit Range	Default & Access	Field Name (ID): Description
31:16	1E0h RO	**Protocol Speed ID Mantissa (PSIM)**
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	0h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	2h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	3h RO	**Protocol Speed ID Value (PSIV)**

16.848 XECP_SUPP_USB3_0 (XECP_SUPP_USB3_0)—Offset 8020h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 3001402h

Bit Range	Default & Access	Field Name (ID): Description
31:24	3h RO	**USB Major Revision: 3.0 (USB3_MAJ_REV)**
23:16	0h RO	**USB Minor Revision (USB_MIN_REV)**
15:8	14h RO	**Next Capability Pointer (NCP)**
7:0	2h RO	**Supported Protocol ID (SPID)**

16.849 XECP_SUPP_USB3_1 (XECP_SUPP_USB3_1)—Offset 8024h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 20425355h

Bit Range	Default & Access	Field Name (ID): Description
31:0	20425355h RO	**XECP_SUPP_USB2_1 (XECP_SUPP_USB2_1):** Namestring USB

16.850 XECP_SUPP_USB3_2 (XECP_SUPP_USB3_2)—Offset 8028h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 7000070Ah

Bit Range	Default & Access	Field Name (ID): Description
31:28	7h RO	**Protocol Speed ID Count (PROT_SPD_ID_CNT):** 1 USB 3.0 Speed (Supper Speed)
27:16	0h RO	**Rsvd0 (Rsvd0)**
15:8	7h RO	**Compatible Port Count (CPC)**
7:0	Ah RO	**Compatible Port Offset (CPO)**

16.851 XECP_SUPP_USB3_3 (XECP_SUPP_USB3_3)—Offset 8030h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 4E00121h

Bit Range	Default & Access	Field Name (ID): Description
31:16	4E0h RO	**Protocol Speed ID Mantissa (PSIM)**
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	1h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	2h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	1h RO	**Protocol Speed ID Value (PSIV)**

16.852 XECP_SUPP_USB3_4 (XECP_SUPP_USB3_4)—Offset 8034h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 9C00122h

Bit Range	Default & Access	Field Name (ID): Description
31:16	9C0h RO	**Protocol Speed ID Mantissa (PSIM)**
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	1h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	2h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	2h RO	**Protocol Speed ID Value (PSIV)**

16.853 XECP_SUPP_USB3_5 (XECP_SUPP_USB3_5)—Offset 8038h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 13800123h

Bit Range	Default & Access	Field Name (ID): Description
31:16	1380h RO	**Protocol Speed ID Mantissa (PSIM)**
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	1h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	2h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	3h RO	**Protocol Speed ID Value (PSIV)**

16.854 XECP_SUPP_USB3_6 (XECP_SUPP_USB3_6)—Offset 803Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 50134h

Bit Range	Default & Access	Field Name (ID): Description
31:16	5h RO	**Protocol Speed ID Mantissa (PSIM)**
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	1h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	3h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	4h RO	**Protocol Speed ID Value (PSIV)**

16.855 XECP_SUPP_USB3_7 (XECP_SUPP_USB3_7)—Offset 8040h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 5B10125h

Bit Range	Default & Access	Field Name (ID): Description
31:16	5B1h RO	**Protocol Speed ID Mantissa (PSIM)**
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	1h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	2h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	5h RO	**Protocol Speed ID Value (PSIV)**

16.856 XECP_SUPP_USB3_8 (XECP_SUPP_USB3_8)—Offset 8044h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: B630126h

Bit Range	Default & Access	Field Name (ID): Description
31:16	B63h RO	**Protocol Speed ID Mantissa (PSIM)**
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	1h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	2h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	6h RO	**Protocol Speed ID Value (PSIV)**

16.857 XECP_SUPP_USB3_9 (XECP_SUPP_USB3_9)—Offset 8048h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 16C60127h

Bit Range	Default & Access	Field Name (ID): Description
31:16	16C6h RO	**Protocol Speed ID Mantissa (PSIM)**
15:9	0h RO	**Rsvd0 (Rsvd0)**
8	1h RO	**PSI Full Duplex (PFD)**
7:6	0h RO	**PSI Type (PLT)**
5:4	2h RO	**Protocol Speed ID Exponent (PSIE)**
3:0	7h RO	**Protocol Speed ID Value (PSIV)**

16.858 Host Controller Capability (HOST_CTRL_CAP_REG)—Offset 8070h

This is a register that describe the host controller the extended cap location. It includes the , XECP_HOST_NEXT_CAP_OFFSET and VEND_DEF_HOST_CAP_ID_192. This register is not subject to HW save and restore.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 4DFFC0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD)**
23:16	4Dh RW/L	**Valid Length (VALID_LENGTH):** Indicates the number of valid DWords in the capability, that need to be saved and restored. 47h -> 818Bh is the last valid byte
15:8	FFh RW/L	**Next Capability Pointer (XECP_HOST_NEXT_CAP_OFFSET)**
7:0	C0h RW/L	**Supported Protocol ID (VEND_DEF_HOST_CAP_ID)**

16.859 Override EP Flow Control (HOST_CLR_MASK_REG)—Offset 8078h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RO	**RESERVED (RSVD)**
9:5	0h WO	**Slot Number Default Config (SNDC):** 5bits of slot number as a default configuration. It can scale to max of 128 slots
4:1	0h WO	**EP Number (EP_NUM):** 4bits of EP number
0	0h WO	**Clear Internal Scheduler's Mask (CISM):** This is a register that is used to clear the internal scheduler's mask that is used to stop scheduling a particular EP. Bit0 indicates the direction of the EP

16.860 Clear Active IN EP ID Control (HOST_CLR_IN_EP_VALID_REG)—Offset 807Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h WO	**Clear Active IN EP ID Control (HOST_CLR_IN_EP_VALID_REG):** This register is used to clear the internal valid IN EP array that TRM stored in order to guarantee one IN EP per port. This register allows software to clear the valid bit of each port IN EP. This field indicates the port number. For a 2port configuration, only bit1:0 are valid. It can scale for the max number of ports that we support.

16.861 Clear Poll Mask Control (HOST_CLR_PMASK_REG)—Offset 8080h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RO	**RESERVED (RSVD)**
9:5	0h WO	**Slot Number Default Config (SNDC):** 5bits of slot number as a default configuration. It can scale to max of 128 slots
4:1	0h WO	**EP Number (EP_NUM):** 4bits of EP number
0	0h WO	**Clear Internal Scheduler's Poll Mask (CISPM):** This is a register that is used to clear the internal scheduler's poll mask that is used to indicate whether we need to poll this EP. This is used for USB2. Bit0 indicates the direction of the EP

16.862 Host Control Scheduler (HOST_CTRL_SCH_REG)—Offset 8094h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 8100h

Bit Range	Default & Access	Field Name (ID): Description
31:21	0h RW	**Scheduler Host Control Reg Cont (SCHED_HOST_CTRL_CONT):** 31:25 Reserved (24): disable marking overlap flag on all TT periodic INs. (23): disable async. scheduling while periodic active to same port (22): disable "level" method of USB2 port periodic done check (on by default) (21): enable "strobe" method of USB2 port periodic done check (off by default)
20:13	4h RW	**TTE Host Control (TTE_HOST_CTRL):** (0): disable interrupt complete split limit to 3 microframes (1): disable checking of missed microframes (2): disable split error request w/NULL pointer on speculative INs with data payload and no TRB. (3): disable deferred split error request on speculative IN with data payload and no TRB. (7:4): reserved
12:11	0h RW	**Cache Size Control Reg (CACHE_SZ_CTRL):** 0: 64 1: 32 2,3: 16
10:9	0h RW	**Maximum EP Per Slot (MAX_EP_SLOT):** 0: 32 1: 16 2: 8 3: 4
8	1h RW	**Turn on scratch_pad_en (TO_SCRATCH_PAD_EN)**

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Scheduler Host Control Reg (SCHED_HOST_CTRL):** (0): disable poll delay (1): disable TRM active in EP valid check (2): enable TTE overlap prevention on interrupt IN EPs (at cost of possible service interval slip) (3) enable TTE overlap prevention on interrupt OUT EPs (at cost of possible service interval slip) (5:4) scheduler sort pattern 00 (default) search ISO ahead of interrupt within each service interval 01 - search USB2-ISO, USB3-ISO, USB2-Interrupt, USB3-Interrupt within each service interval 10 - search strictly by interval 11 - search all ISO intervals ahead interrupt intervals and within each interval, USB2 ahead of USB3 (6): disable 1 pack scheduling limit when ISO pending in present microframe (7): enable check to stop scheduling on port that are not connected

16.863 Global Port Control (HOST_CTRL_PORT_CTRL)—Offset 80A0h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 380Fh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**OVERFLOW ERROR DETECTION ENABLE (OVERFLOW_ERR_DETECT_EN)**
30:12	3h RW	**HBUF_WATER_MARK (HBUF_WATER_MARK)**
11	1h RW	**CPL Cut Thru Enable (CPL_CUT_THRU_EN)**
10:4	0h RW	**RESERVED (RSVD0)**
3:0	Fh RW	**RESERVED (RSVD1)**

16.864 Power Management Control (PMCTRL_REG)—Offset 80A4h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 2DFF90h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Async PME Source Enable (ASYNC_PME_SRC_EN):** This field allows the async PME source to be allowed to generate PME. This is specifically required for SOCs that do not allow for any clock other than RTC to be available during RTD3.
30	0h RW	**Legacy PME Source Enable (LEGACY_PME_SRC_EN):** This field allows the legacy PME source to be used in PME generation. The legacy source in in reference to the source prior to the RTD3 changes.
29	0h RW	**Reset Warn Power Gate Trigger Disable (RESET_WARN_PWR_GATE_TRIGGER_DISABLE):** This field controls the actions taken for due to reset warn. 0 - Reset Warn will trigger a HW autonomous Power Gate 1 - Reset Warn will not trigger a HW autonomous Power Gate
28	0h RW	**CLR_PME_FLAG_PULSE_AUX_CCLK (CLR_PME_FLAG_PULSE_AUX_CCLK)**
27	0h RW	**RESERVED (RSVD)**
26	0h RW	**XLFPSCOUNTSRC (XLFPSCOUNTSRC):** XLFPSCOUNTSRC (Source for LFPS OFF Counter) 0: Central RTC Counter for LFPS detection 1: Local Counter for LFPS detection
25	0h RW	**XELFPSRTC (XELFPSRTC):** XELFPSRTC (Enable LFPS Filtering on RTC) 0: Use Oscillator clock for LFPS Filtering during P3 1: Use RTC Clock for LFPS Filtering during P3
24	0h RW	**XMPHYSPGDD0I2 (XMPHYSPGDD0I2):** XMPHYSPGDD0I2 (ModPhy Sus Well Power Gate Disable for D0I2) 0 : Modphy sus well power gating enabled 1 : Modphy sus well power gating disabled
23	0h RW	**XMPHYSPGDD0I3 (XMPHYSPGDD0I3):** XMPHYSPGDD0I3 (ModPhy Sus Well Power Gate Disable for D0I3) 0 : Modphy sus well power gating enabled 1 : Modphy sus well power gating disabled
22	0h RW	**XMPHYSPGDRTD3 (XMPHYSPGDRTD3):** XMPHYSPGDRTD3 (ModPhy Sus Well Power Gate Disable for RTD3) 0 : Modphy sus well power gating enabled 1 : Modphy sus well power gating disabled
21:18	Bh RW	**XD3RTCPTTM (XD3RTCPTTM):** XD3RTCPTTM (D3 RTC Port Timer Tick Multiplier) This register will be the multiplication factor for determining SSIC Wake Detection Frequency and RXDET based on the XD3RTCPTTC value. If XD3RTCPTTC is 9h and this register is Bh, frequency for RXDET and SSIC H8EXIT detection while MODPHY SUS Power gating is enabled would be 99ms.
17	0h RW	**U3 LFPS Periodic Sampling ON Time Control (U3_LFPS_PRDC_SAMPLING_ON_TIME_CTRL):** This field controls the ON time for the LFPS periodic sampling for USB3/SSIC ports. 0 ON time is 2 rtc clocks 1 ON time is 3 rtc clocks Note: This field is ignored if USB3/SSIC PHY SUS Well Power Gating is enabled.

Bit Range	Default & Access	Field Name (ID): Description
16	1h RW	**AON LFPS Detector Enable Mode (AON_LFPS_DETECTOR_EN_MODE):** 1 - Allow the LFSP Detector in AON to own LFPS detection when the port is in PS3 for U2/U3 - not RxD regardless of port ownership. 0 - Allow the LFPS Detector in AON to own the LFPS detection only when the AON owns the port and in U2/U3 - not RxD
15:8	FFh RW	**SS U3 LFPS Detection Threshold (SS_U3_LFPS_DETECTION_THRESHOLD):** This field controls the threshold used to determine when a valid U3 Wake is detected through when using the unfiltered LFPS source. The value on this field will reflect the binary count required to have been detected on the counter being clocked by the unfiltered lfps source to result in a valid U3 wake detection.
7:4	9h RW	**SS_U3_LFPS_PRDC_SAMPLING_OFFTIME_CTRL (SS_U3_LFPS_PRDC_SAMPLING_OFFTIME_CTRL):** This field controls the OFF time for the LFPS periodic sampling for USB3 Ports 0x0 periodic sampling is disabled. 0x1 OFF time is 1ms 0x2 OFF time is 2ms 0xF OFF time is 15ms The ON Time is determined by the amount of time required to reliably determine if there is a valid LFPS and is HW implementation specific. A speed up mode shall be implemented where this field is in units of us. i.e. 0x1 = 1 us OFF time, 0x2 = 2 us OFF time, etc.
3	0h RW	**PS3 LFPS Source Select (PS3_LFPS_SRC_SEL):** 0 LFPS Source is unfiltered 1 LFPS Source is filtered (Rx-Elec-Idle) LFPS Source is Rx-Elec-Idle for any non PS3 state.
2	0h RW	**XHCI Engine Autonomous Power Gate Exit Reset Policy (XHC_AUTO_PWRGATE_EXITRST_POLICY):** Controls when the xHCI engine is brought out of reset due to a power ungate. 0 Engine is brought out of reset when D3 to D0 is triggered. This allows for a quick power up sequence while leaving the virtual PCIe LTSSM in L23 is power ungate is not due to D3 to D0. 1 Engine is brought out of reset along with the rest of the IP. This is required for PMC save/restore flow.
1	0h RW	**USB2 Port Wake Unit Coupling Policy (USB2_PORT_WAKE_COUPLING_POLICY):** Controls the trigger for USB2 Port Wake Units to initiate Port Level Power Off Preparation. 0 RTD3 triggered 1 - Port Triggered when in L1, L2 or Disabled, Disconnected
0	0h RW	**USB3 Port Wake Unit Coupling Policy (USB3_PORT_WAKE_COUPLING_POLICY):** Controls the trigger for USB3 Port Wake Units to initiate Port Level Power Off Preparation. 0 - RTD3 Triggered 1 - Port Triggered when in PS3 due to RxDetect, U3, U2 or Disabled

16.865 PGCB Control (PGCBCTRL_REG)—Offset 80A8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 315555h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**RESERVED (RSVD)**
27:25	0h RW	**IP_INACCESSIBLE_HYSTERESIS_TIMER (IP_INACCESSIBLE_HYSTERESIS_TIMER):** 000 ? 50us 001 ? 100us 010 ? 150us 011 ? 200us 000 ? 250us 001 ? 300us 010 ? 350us 111 ? 400us
24	0h RW	**Override_Disable (OVERRIDE_DISABLE)**
23:20	3h RW	**CDH Aggregation Minimum Wait Time (CDH_MIN_WAIT_TIME):** This field controls the minimum time that we must wait for resets to propagate before allowing Power Up flow to complete. The Power Up flow requires special handling of clocks, reset and sleep signaling which requires the CDH to monitor reset propagation. This timer allow for flexibility on when we consider all resets propagated. 0h Disabled 1h 2 clocks 2h 4 clocks 3h 8 clocks 4h 16 clocks 5h 32 clocks 6h 64 clocks 7h 128 clocks
19	0h RW	**RESERVED (CDH_RST_PROPAGATION_CNTRL):** This bit will indicate the mode of operation for the CDH Reset Propagation Aggregator. It allows for a mode where the aggregation will wait for an indication from each CDH before proceeding OR ignore the CDH indication before proceeding. Regardless of the mode, the aggregator will enforce the CDH Aggregation Mini Wait Time. 0 Aggregate all the CDH indications before triggering the CDH Aggregation Mini Wait Time. Once the Min Wait Time is met a notification to the Power Sequencer/Control will be initiated 1 Trigger the CDH Aggregation Mini Wait Time w/o waiting for CDH indication. Once the Min Wait time is met a notification to the Power Sequencer/Control will be initiated.
18	0h RW	**MMP_PFET_REQ_OVRD (MMP_PFET_REQ_OVRD):** This bit will disable the MMP PFET request condition for PGCB control. 1 MMP PFET Request Ignored 0 Default
17:16	1h RW	**PGCB Clock Gate Request to PGCB Sleep (PGCB_CLK_GATE_REQ_2_PGCB_SLEEP):** Value representing the minimum number of delay clocks required between the assertion of pgcb_ip_clkgate_b to the deassertion of pgcb_sleep on PG exit. (This is only applicable for IP-Accessible PG with state-retention enabled.) Please see the description of cfg_tsleepact for more details.

Bit Range	Default & Access	Field Name (ID): Description
15:14	1h RW	**PGCB Sleep Deassertion to PGCB ISM Unlock Req (PGCB_SLEEP_DEASRTN_2_ISM_UNLOCK_REQ):** For IP-Accessible PG (with state retention): Value representing the minimum number of delay clocks required between the deassertion of gcb_sleepto the deassertion of pgcb_ip_*_lock_req_b. For IP-Inaccessible PG (or with state-retention disabled): Value representing the minimum number of delay clocks required between the deassertion of pgcb_sleep to the deassertion of pgcb_isol_en_b.
13:12	1h RW	**PGCB Prim Reset Deassertion to PGCB Next State (PGCB_PRIMRST_DEASRTN_2_NSTATE):** For IP-Accessible PG (with state retention): Value representing the minimum number of delay clocks required between the deassertion of pgcb_sleep to the deassertion of pgcb_ip_*_lock_req_b. For IP-Inaccessible PG (or with state-retention disabled): Value representing the minimum number of delay clocks required between the deassertion of pgcb_sleep to the deassertion of pgcb_isol_en_b.
11:10	1h RW	**PGCB Side Reset Deassertion to PGCB Prim Reset Deassertion (PGCB_SIDERST_DEASRTN_2_PRIMERST_DEASRTN):** Value representing the number of delay clocks required between the deassertion of gcb_force_rst_bto the deassertion of gcb_force_prim_rst_b (This is only applicable for IP-Accessible PG.) Please see the description of cfg_tsleepact for more details.
9:8	1h RW	**PGCB Latch Isolation High to PGCB Clock Ungate Request (PGCB_LATCH_ISO_HI_2_PGCB_CLK_UNGATE_REQ):** Value representing the number of delay clocks required between the assertion of pgcb_isol_latchen to the deassertion of pgcb_ip_clkgate_req_b. Please see the description of cfg_tsleepact for more details.
7:6	1h RW	**PGCB Isolation Deassertion to PGCB Latch Isolation High Count (PGCB_ISO_DEASRTN_2_PGCB_ISO_HI_CNT):** Value representing the minimum number of delay clocks required between the deassertion of pgcb_isol_en_b to the assertion of gcb_isol_latchen. Please see the description of cfg_tsleepact for more details.
5:4	1h RW	**PGCB Reset Assertion to PGCB Power Down Request Count (PGCB_RST_2_PGCB_PWRDOWN_REQCNT):** For IP-Accessible PG (with state retention): Value representing the minimum number of delay clocks required between the assertion of pgcb_force_prim_rst_b and pgcb_force_rst_b to the assertion of pgcb_pmc_pg_req_b. For IP-Inaccessible PG (or with state-retention disabled): Value representing the minimum number of delay clocks required between the assertion of pgcb_force_prim_rst_b and pgcb_force_rst_b to the assertion of pgcb_sleep. Please see the description of cfg_tsleepact for more details.

Bit Range	Default & Access	Field Name (ID): Description
3:2	1h RW	**PGCB Isolation Assertion to PGCB Reset Assertion Count (PGCB_ISO_ASRTN_2_PGCB_RST_ASRTN_CNT):** Value representing the minimum number of delay clocks required between the assertion of gcb_isol_en_bto the assertion of gcb_force_prim_rst_band gcb_force_rst_b Please see the description of [1:0] for more details.
1:0	1h RW	**PGCB Sleep Assertion to PGCB Isolation Enable Count (PGCB_SLEEP_ASRTN_2_PGCB_ISO_EN_CNT):** For IP-Accessible PG (with state retention): Value representing the minimum number of delay clocks required between the assertion of pgcb_sleep (and the deassertion of pgcb_isol_latchen) to the assertion of pgcb_isol_en_b. For IP-Inaccessible PG (or with state-retention disabled): Value representing the minimum number of delay clocks required between the deassertion of pgcb_isol_latchen to the assertion of pgcb_isol_en_b, as well as the minimum number of delay clocks required between the assertion of pgcb_sleep to the assertion of pgcb_pmc_pg_req_b. Common for all cfg_t* inputs: 0 1 clock 1 2 clocks 0 8 clocks 1 256 clocks For any of the cfg_t* inputs that an IP may feel are a don-care and the IP does not feel they will be useful for post-silicon debug, the recommendation is to tie the input to 201 (2 clocks). Otherwise, the inputs may be tied to another relevant value or connected to a configuration register. Note: These signals may only change while pgcb_pwrupidle is asserted, if pgcb_pwrupidle is deasserted it should be valid and stable. (It may change the same cycle that ip_pgcb_pg_rdy_req_b asserts.)

16.866 D0I3 Control (DOI3CTRL_REG)—Offset 80ACh

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 8h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**RESERVED (RSVD)**
3	1h RW/1C	**RestoreRequired (RESTORE_REQUIRED):** When set (by HW), SW must restore state to the IP. The state may have been lost due to a reset or full power lost. SW clears the bit by writing a 1. This bit will be set on initial power up.
2	0h RW	**D0I3 (D0I3):** SW sets this bit to 1 to move the IP into the D0i3 state. Writing this bit to 0 will return the IP to the fully active D0 state (D0i0).
1:0	0h RO	**RESERVED (RSVD1)**

16.867 HOST_CTRL_MISC_REG (HOST_CTRL_MISC_REG)—Offset 80B0h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 1037Fh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**USB2_LTRUPDT_DIS (USB2_LTRUPDT_DIS)**
30	0h RW	**USB2 Line State Debounce During Port Reset Policy (USB2_LINE_STATE_DEBOUNCE_DURING_PORT_RESET_POLICY):** This register controls how the debounce is enforced during the Port Reset phase. 0 do not enable the line state debounce during port reset. 1 enable the line state debounce during port reset.
29	0h RW	**TTE PEXE Credit Fix Disable (TTE_PEXE_CREDIT_FIX_DISABLE):** When set, it disables a fix implemented to re-deem PEXE credits when a port is disconnected
28	0h RW	**TTE Scheduling policy (TTE_SCHEDULING_POLICY):** This register controls a fix made to prevent over-scheduling by not account for 188B in each uFrame. Setting this bit will disable the fix and allow for over-scheduling.
27	0h RW	**USB3 ITP Delta Timer Source Select (USB3_ITP_DELTA_TIMER_SOURCE_SELECT):** This register selects the source for the delta timer tracking used for ITP generation. 0 the source is a 16.666 ns tick generated from a crystal reference clock 1 - the source is a 16.666 ns tick generated from the aux_cclk. This field needs to remain in sync with Frame Timer Source Select to ensure the are both set or both cleared. There is no support for any other combination.
26	0h RW	**Frame Timer Source Select (FRAME_TIMER_SOURCE_SELECT):** This register controls the source for the frame timer. 0 the source for the frame timer is a crystal reference clock 1 the source for the frame timer is the aux_cclk.
25	0h RW	**uFrame Masking Enable (UFRAME_MASKING_ENABLE):** If set, enables the uFrame tick to be masked due to ports being in U3/NC. This controls a fix made to disable the auto masking of uFrame tick due to port state w/o any pipeline idle condition. When cleared, we rely on gating of frame timer due to proper port state and idleness tracking from the pipeline.
24	0h RW	**Late FID Check Disable (LATE_FID_CHECK_DISABLE):** This register disables the Late FID Check performed when starting an ISOCH stream.

Bit Range	Default & Access	Field Name (ID): Description
23:20	0h RW	**RESERVED (RSVD1)**
19	0h RW	**USB2 Resume Cx Inhibit Disable (USB2_RESUME_CX_INHIBIT_DISABLE):** Controls if USB2 L1 Resume is allowed to contribute to DMA Active which will inhibit Cx state. 0 USB2 L1 Resume is allowed to inhibit Cx via DMA Active 1 USB2 L1 Resume is NOT allowed to inhibit Cx via DMA Active When cleared, Cx will only be inhibited when the DMA traffic for the port begins.
18:16	1h RW	**Extra uFrame (EXTRA_UFRAME):** This register controls the extra number of uFrames added onto the advancing of late FID check.
15:0	37Fh RW	**Valid Isoch Scheduling Range (VALID_ISOCH_SCHEDULING_RANGE):** This register defines the window in miliseconds from the current Frame that will be considered for scheduling in an upcoming Frame. Anything scheduled outside of this window will be considered as late and will trigger the Missed Service Error.

16.868 HOST_CTRL_MISC_REG2 (HOST_CTRL_MISC_REG2)—Offset 80B4h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RW	**max_short_pkt_adv_cnt (MAX_SHORT_PKT_ADV_CNT)**
28	0h RW	**dis_sch_frameid_chk (DIS_SCH_FRAMEID_CHK)**
27:13	0h RW	**misc_config_reg2_27_13_rsvd (RSVD1)**
12	0h RW	**disable_idt_fix_odma (DISABLE_IDT_FIX_ODMA):** arc fix 0:enable the fix 1:disable the fix
11	0h RW	**disable_ping_fix_odma (DISABLE_PING_FIX_ODMA):** fix 0:enable the fix 1:disable the fix
10	0h RW	**disable_cerr_fix_idma 0: fix is enabled 1:fix is disabled (DISABLE_CERR_FIX_IDMA):**
9	0h RW	**en_100ms_watch_dog_timer (EN_100MS_WATCH_DOG_TIMER)**
8	0h RW	**en_watch_dog_timer (EN_WATCH_DOG_TIMER)**

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**misc_config_reg2_bit7_rsvd (RSVD2)**
6	0h RW	**disable_tcg_ungate_on_flush (DISABLE_TCG_UNGATE_ON_FLUSH)**
5	0h RW	**disable_vnn_frame_timer (DISABLE_VNN_FRAME_TIMER)**
4	0h RW	**disable_clr_ccs_on_cas_set (DISABLE_CLR_CCS_ON_CAS_SET)**
3	0h RW	**disable_rhub_park_at_dbcDisc (DISABLE_RHUB_PARK_AT_DBCDISC)**
2	0h RW	**disable_block_wpr_on_disPorts (DISABLE_BLOCK_WPR_ON_DISPORTS)**
1:0	0h RW	**HOST_CTRL_MISC_REG2_bits1_0 (HOST_CTRL_MISC_REG2_1_0)**

16.869 SSPE_REG (SSPE_REG)—Offset 80B8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**ssCfgBlockPwrDwn4ActLFPS (SS_CFG_BLOCK_PWRDWN_4_ACT_LFPS)**
30	0h RW	**dis_clr_ccs_4hcreset (DIS_CLR_CCS_4_HCRESET)**
29	0h RW	**disable_rawlfps_based_wake_fix (DISABLE_RAWLFPS_BASED_WAKE_FIX)**
28:7	0h RO	**Rsvd (Rsvd)**
6:0	0h RW	**SSPE_REG (SSPE_REG)**

16.870 (SSPITPE)—Offset 80BCh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 7Fh

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**RSVD (RSVD)**
6:0	7Fh RW	**ITP Transmit Enable (ITP_TRANSMIT_EN):** Width is scaled with USB3 Port Count PortCount=4

16.871 AUX Reset Control (AUX_CTRL_REG)—Offset 80C0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 15FC0F0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Reset USB3 AUX/Main PD Logic (RST_U3_AM_PDL):** A reset that is designed to reset all USB3 port AUX + Main power domain logic. This is a debug function. It is called cold reset. Write 1 to this register will issue a cold reset to USB3 ports.
30	0h RW	**RESERVED (RSVD)**
29	0h RW	**Enable PCIe PIPE Reset As PCIe PHY Reset (EN_PPIPE_PPHY_RST):** This bit set to 1 selects the PCIe PIPE reset as a PCIe PHY power on reset. 0 selects the AUX powerup reset as a PCIe PHY powerup reset.
28	0h RW	**Ensure PIPE Powerdown On PERST# (EN_PPWRDN_PERST):** This bit ensures any PERST# being asserted will cause PIPE powerdown state transition to P1.
27	0h RW	**Disable PERST# Main RST (DIS_PERST_MRST):** There is a feature where we only allow PERST# to be treated as a main powerdown reset when it is asserted during the state that is not in D3. This bit disables this feature when it is set to 1. If this bit is set to 1, it means that any PERST# will be treated as a main power domain reset regardless of its Dstate.
26	0h RW	**Disable PCIe PERST# Host Controller (DIS_PERST_HC):** There is a feature where we can disable PCIe PERST# from reset the host controller until power management control module has done its clock switching. This is for a case where PM entered L23 and immediately PERST# asserted. This bit is designed to enable this feature when set to 1.
25	0h RW	**Enable Fast Simulation Reset Mode (EN_FAST_RST):** Enable a fast simulation mode for reset function. This is a feature for GLS
24	1h RW	**Prevent USB3 Link Down Reset (PREV_U3_LDRST):** When set to 1 prevent a reset being generated due to the USB3 port link down condition.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**Ignore Main Power Up Reset (IGN_MPU_RST):** When set to 1 ignore the main power up reset for USB3 PIPE PHY reset.
22	1h RW	**RESERVED (RSVD_1)**
21	0h RW	**Ignore HC Reset USB2 (IGN_HC_RST_U2):** When set to '1' ignore HC reset to reset the USB2 Port logic
20	1h RW	**Ignore HC Reset USB PHY (IGN_HC_RST_UP_POR):** When set to '1' ignore HC reset to the USB PHY power-on reset
19	1h RW	**Enable PCIe Link-Down Reset (EN_PLD_RST):** Enable a reset due to a PCIe link-down condition. The PCIe link down condition will cause a HC reset liked. If this bit is set 1, the PCIe link down condition will only reset the PCIe core.
18	1h RW	**Enable EEPROM Reload On Power Up (EN_EEP_REL_PU):** When set to '1' enable EEPROM reload on every main power-up
17	1h RW	**Ignore HC Reset PCIe PHY PIPE (IGN_HC_RST_PPP):** When set to '1' ignore HC reset to the PCIe PHY PIPE reset
16	1h RW	**Ignore LTSSM Reset USB PHY PIPE (IGN_LRST_UPP):** When set to '1' ignore the LTSSM of USB link state transition caused reset to USB PHY PIPE reset
15	1h RW	**Ignore Warm Reset USB PHY Power (IGN_WR_UPP):** When set to '1' ignore warm reset the USB PHY power on reset
14	1h RW	**Allow Core PCIe Link Down Reset (ALL_CPLD_RST):** When set to '1' allow PCIe link down to cause a reset to the rest of the core
13	0h RW	**Ignore Hot Reset USB3 (IGN_HR_U3):** When set to '1' ignore hot reset to the USB3 port logic
12	0h RW	**Ignore Warm Reset USB3 (IGN_WR_U3):** When set to '1' ignore warm reset to the USB3 port logic
11	0h RW	**Ignore Main Power Up Reset USB3 (IGN_MPU_RST_U3):** When set to '1' ignore main power up reset to USB3 port logic
10	0h RW	**Ignore Main Power Up Reset USB2 (IGN_MPU_RST_U2):** When set to '1' ignore main power up reset to USB2 port logic
9	0h RW	**Ignore Main Power Up Reset PCIe Core (IGN_MPU_RST_PC):** When set to '1' ignore main power up reset to PCIe core
8	0h RW	**Ignore Main Power Up Reset PCIe PHY (IGN_MPU_RST_PP):** When set to '1' ignore main power up reset to PCIe PHY
7	1h RW	**Ignore HC Reset USB PHY (IGN_HC_RST_UP):** When set to '1' ignore HC reset to the USB PHY
6	1h RW	**Ignore Warm Reset USB PHY (IGN_WRST_UP):** When set to '1' ignore warm reset to the USB PHY
5	1h RW	**Enable HC Reset Per Port Isolation (EN_HC_RST_PPI):** Enables the HC reset or per port reset isolation function

Bit Range	Default & Access	Field Name (ID): Description
4	1h RW	**Allow Power Off Power Domain Reset (ALL_PO_PDRST):** When set to '1' allow main power off condition to trigger a main power domain reset
3	0h RW	**Ignore Wait For PERST# During Power Show Down (IGN_PERST_PSD):** When set to '1' ignore waiting for PERST# deassertion during main power show down.
2	0h RW	**Ignore Fundamental Reset During AUX Power Up (IGN_FRST_AUX_PU):** When fundamental reset is asserted during AUX power up, if this bit is set, then we will ignore PERST# such that purely wait for timeout to deassert fundamental reset.
1:0	0h RW	**Trigger Fundamental Reset (TRIG_FRST):** Writing to bit(1:0) to value of 2'b11 will cause a fundamental reset

16.872 Super Speed Bandwidth Overload (HOST_BW_OV_SS_REG)—Offset 80C4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 4A4008h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**RESERVED (RSVD)**
23:12	4A4h RW	**Max. TT BW Allowed (MAX_TT_BWA):** see white paper
11:0	8h RW	**Per Packet Overhead SS BW (PP_OVRH_SSBW):** BW calculation: Overhead per packet for SS BW calculations. see white paper.

16.873 High Speed TT Bandwidth Overload (HOST_BW_OV_HS_REG)—Offset 80C8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 1A01Fh

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**RESERVED (RSVD)**

Bit Range	Default & Access	Field Name (ID): Description
23:12	1Ah RW	**Per Packet Overhead HS-TT BW (PP_OVRH_HSTTBW):** BW calculation: Overhead per packet for HS-TT BW calculations. see white paper.
11:0	1Fh RW	**Per Packet Overhead HS BW (PP_OVRH_HSBW):** BW calculation: Overhead per packet for HS BW calculations. see white paper.

16.874 Bandwidth Overload Full Low Speed (HOST_BW_OV_FS_LS_REG)—Offset 80CCh

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 14080h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**RESERVED (RSVD)**
23:12	14h RW	**Per Packet Overhead FS BW (PP_OVRH_FSBW):** BW calculation: Overhead per packet for FS BW calculations. see white paper.
11:0	80h RW	**Per Packet Overhead LS BW (PP_OVRH_LSBW):** BW calculation: Overhead per packet for LS BW calculations. see white paper.

16.875 System Bandwidth Overload (HOST_BW_OV_SYS_REG)—Offset 80D0h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 32010h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**RESERVED (RSVD)**
23:12	32h RW	**Per TT Packet Overhead System BW (PTTP_OVRH_SBW):** BW calculation: Overhead per TT packet for System BW calculations. see white paper.

Bit Range	Default & Access	Field Name (ID): Description
11:0	10h RW	**Per Packet Overhead System BW (PP_OVRH_SBW):** BW calculation: Overhead per packet for System BW calculations. see white paper.

16.876 Scheduler Async Delay (HOST_CTRL_SCH_ASYNC_DELAY_REG)—Offset 80D4h

Global defaults for inserting delays between packets in the scheduler for async. types.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RO	**RESERVED (RSVD)**
19	0h RW	**High-Speed Bulk Delay Enable (HS_BD_EN)**
18:16	0h RW	**High-Speed Bulk Delay Default (HS_BD_DEF):** (0=125us,1=250us,2=500us,3=1ms,...)
15	0h RW	**Full-Speed Bulk Delay Enable (FS_BD_EN)**
14:12	0h RW	**Full-Speed Bulk Delay Default (FS_BD_DEF):** (0=125us,1=250us,2=500us,3=1ms,...)
11	0h RW	**High-Speed Control Delay Enable (HS_CD_EN)**
10:8	0h RW	**High-Speed Control Delay Default (HS_CD_DEF):** (0=125us,1=250us,2=500us,3=1ms,...)
7	0h RW	**Full-Speed Control Delay Enable (FS_CD_EN)**
6:4	0h RW	**Full-Speed Control Default (FS_CD_DEF):** (0=125us,1=250us,2=500us,3=1ms,...)
3	0h RW	**Low-Speed Control Delay Enable (LS_CD_EN)**
2:0	0h RW	**Low-Speed Control Delay Default (LS_CD_DEF):** (0=125us,1=250us,2=500us,3=1ms,...)

16.877 DEVICE MODE CONTROL REG 0 (DUAL_ROLE_CFG_REG0)—Offset 80D8h

All bits in this register must be in the Always ON Power domain (ungated SUS or AON as appropriate)

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 800h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**XDCI_DRD_CTRL_EN (REG_IN_XDCI):** When DRD_ACCESS_MODE (bit 18) =1, 0 -- The policies below is controlled by register in the XHCI MMIO 1 -- The policies below is controlled by register in the XDCI MMIO When DRD_ACCESS_MODE =0, this bit is a don?t care.
30:28	0h RW	**DRD_WDT_TIMER (DRD_WDT_TIMER):** 0-based incremental of 500ms
27	0h RW	**EN_DRD_WDT_SWITCH_FROM_HOST_TO_DEVICE (EN_DRD_WDT_SWITCH_FROM_HOST_TO_DEVICE)**
26	0h RW	**EN_DRD_WDT_SWITCH_FROM_DEVICE_TO_HOST (EN_DRD_WDT_SWITCH_FROM_DEVICE_TO_HOST)**
25	0h RW	**EN_DRD_WDT_SWITCH_OUT_OF_COLD_RESET (EN_DRD_WDT_SWITCH_OUT_OF_COLD_RESET)**
24	0h RW	**SW_VBUS_VALID (SW_VBUS_VALID):** if SW_IDPIN_EN (bit 21) is 1, 0 ? deassert sw vbus valid 1 ? assert sw vbus valid
23	0h RW	**EN_PIPE_4_1_SYNC_PHY_STATUS (EN_PIPE_4_1_SYNC_PHY_STATUS)**
22	0h RW	**EN_PIPE_RX_ON_IDPIN (EN_PIPE_RX_ON_IDPIN):** During the connection to a device, there may be a delay in DRD switch from XDCI to XHCI mode,the rx term can be low after idpin deasserts. If this bit is 0, the rx term will be assert immediately after idpin toggle. Otherwise, the device may fall back to USB2 mode. 1 -- drive 0s on utmi rx signals to controller if not connected.
21	0h RW	**SW_IDPIN_EN (SW_IDPIN_EN):** SW_IDPIN_EN 1 -- enable SW id pin (pre DRD) 0 -- disable SW id pin.
20	0h RW	**SW_IDPIN (SW_IDPIN):** if SW_IDPIN_EN (bit 21) is 1, 0 - host mode 1 - device mode
19	0h RW	**USB2_SUSP_OR_DIS (USB2_SUSP_OR_DIS):** 1 - The DRD UTMI suspendm will be controlled by host/device based on the DRD switch. 0 - whenever device or host deassert suspendm, the DRD UTMI suspendm will be deasserted.
18	0h RW	**DRD_ACCESS_MODE (DRD_ACCESS_MODE):** 0 - The DUAL_ROLE register from host and device are ORed to control DRD 1 - The DUAL_ROLE register from host and device are selected by XDCI_DRD_CTRL_EN (bit 31) to control DRD
17	0h RW	**EN_DIRECT_DRD_SWITCH_ON_USB3PORT_BY_IDPIN (EN_DIRECT_DRD_SWITCH_ON_USB3PORT_BY_IDPIN):** 0 enable the direct DRD switch on USB3 port by idpin 1 disable the direct DRD switch

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW	**SW_SWITCH_ENABLE (SW_SWITCH_ENABLE):** SW switch enable 0 (default) ID pin HW controlled DRD. 1 -- SW controlled DRD (ignore idpin), switch based on the DRD_CONFIG.
15	0h RW	**RSVD (RSVD)**
14:3	100h RW	**DEBOUNCE_VAL (DEBOUNCE_VAL):** ID ping debounce timer (DEBOUNCE_VAL): in the unit of RTC clock (33us) default to 8.448 ms
2	0h RW	**SYNCHRONIZE_SS_HS_SWITCH (SYNCHRONIZE_SS_HS_SWITCH):** Synchronize the SS and HS switch: 0 (default) Does not synchronize. i.e. HS switch on the debounced id pin, while SS switch independently controlled by the sequencer. 1 synchronize HS and SS switch. Both speeds switch when sequencer switch.
1:0	0h RW	**DRD_CONFIG (DRD_CONFIG):** 00 Dynamic DRD switch mode 01 static host mode 10 static device mode 11 -- reserved

16.878 DEVICE MODE CONTROL REG 1 (DUAL_ROLE_CFG_REG1)—Offset 80DCh

All bits in this register must be in the Always ON Power domain (ungated SUS or AON as appropriate)

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 10000000h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**BVALID_HW (HW_BVALID):** hardware bvalid
30	0h RO	**DRD_STATUS_B31_30 (DRD_STATUS_30)**
29	0h RO	**DRD_MODE (DRD_MODE):** SS DRD mode 0 device mode 1 host mode
28	1h RO	**IDPIN (IDPIN):** IDPIN value hw version
27:16	0h RO	**DRD_STATUS_B27_16 (DRD_STATUS_B27_16)**
15:0	0h RW	**DRD_CONTROL_B15_0 (DRD_CONTROL_B15_0)**

16.879 AUX Power Management Control (AUX_CTRL_REG1)—Offset 80E0h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 808DBCA0h

Bit Range	Default & Access	Field Name (ID): Description
31	1h RW	**D3 Hot function enable register (D3_HOT_FXN_EN):** This bit is from pin input which is set 1. But we allow software to alter it if it is needed. 1: D3 hot enabled 0: D3 hot not abled.
30	0h RW	**Allow L1 Core Clock Gating (ALL_L1_CORE_CG):** When set to 1 allows core clock being gated during L1 state.
29	0h RW	**Allow Engine PHY Status Extension (AL_EP_SEXT):** When set to 1 allows the engine to extend PHY status of PCIe PIPE for one more cycle. This is due to the fact that our rate change function has a potential of not being able to sample the phystatus signal.
28	0h RW	**Allow Engine PCIe Rate Change Passing (ALL_EP_RCP):** When set to 1 allows the engine to pass PCIe rate change signal as it is from PCIe core to PCIe PHY.
27	0h RW	**Allow Engine PERST Fundamental Reset (AL_PERST_FRST):** When set to 1 allow engine to treat PERST# as a foundamental reset
26	0h RW	**Overwrite PCIe P2 to P1 (OVR_PCIE_P2_P1):** When set to 1 will overwrite a PCIe powerdown state of P2 to P1.
25	0h RW	**Set Internal SSV 1 (SET_ISSV_1):** When set to 1 set the internal SSV to 1.
24	0h RW	**Clear Internal SSV 0 (CLR_ISSV_0):** When set to 1 clear the internal SSV to 0.
23	1h RW	**Enable save_restore_enable SW Loading (EN_SRE_SW_LD):** This is a bit that enables the save_restore_enable signal being loaded when a software command has set Save bit. This is a debug function.
22	0h RW	**RESERVED (RSVD_1)**
21	0h RW	**Force save_restore 1 (FORCE_SR1):** When set to 1, it will force the save_restore flag to 1. This flag is an bit to ensure that we have masked the update during low power state. If software write this bit to 1, it must write it to 0 in order to resume the normal save and restore function.
20	0h RW	**CFG DISABLE_WARM_RST_DET_specUpPorts (cfg_dis_WrstDet_specU)**
19	1h RW	**cfg iob drivestrength[1] (CIDS1)**

Bit Range	Default & Access	Field Name (ID): Description
18	1h RW	**cfg iob drivestrength[0] (CIDS0)**
17	0h RW	**Enable CFG USB P2 (EN_CFG_UP2):** When set to '1' enable cfg usb p2
16	1h RW	**cfg clk gate dis (CCGD)**
15	1h RW	**Enable CFG RXDET P3 (EN_CFG_RDP3):** When set to '1' enable cfg rxdet p3
14	0h RW	**Enable CFG PIPE Reset (EN_CFG_PIPE_RST):** When set to '1' enable cfg pipe rst
13	1h RW	**Enable Filter TX Idle (EN_FILT_TX_IDLE):** When set to 1 enables a filter function to TX electrical idle signal at PCIe PIPE. We have a filter that will set TXelecidle signal of PCIe PIPE to 1 whenever we are in isolation state or power down transition states.
12	1h RW	**Enable Host Engine Generate PME (EN_HE_GEN_PME):** This is a global switch to whether or not eable this host engine to generate PME message.
11	1h RW	**Enable Isolation (EN_ISOL):** When set to '1' enable isolation
10	1h RW	**Enable L1 Caused P2 Overwrite (EN_L1_P2_OVR):** Set 1 to enable a new feature. This new feature is designed to use L1 as a state to identify whether we should do P2 Overwrite or not. We used to use P1 state to identify whether or not to invoke P2 overwrite function.
9	0h RW	**Enable Core Clock Gating (EN_CORE_CG):** When set to '1' enable core clock gating based on low power state entered
8	0h RW	**Enable PHY Status Timeout (EN_PHY_STS_TO):** When set to '1' enable PHY status timeout function which is designed to cover the PCIePHY issue that we may have not able to detect the PHY status toggle.
7	1h RW	**Ignore aux_pm_en PCIe Core (IGN_APE_PC):** When set to '1' ignore the aux_pm_en reg from PCIe core to continue the remote wake/clock switching support
6	0h RW	**Enable P2 Overwrite P1 (EN_P2_OVR_P1):** When set to '1' enable P2 overwrite P1 when PCIe core has indicated the transition from P0 to P1. This is to enable entering the even lower power state.
5	1h RW	**Enable P2 Remote Wake (EN_P2_REM_WAKE):** When set 1 '1' enable the remote wake function by allowing P2 clock/switching and P2 entering
4:1	0h RW	**Forced PM State (FORCED_PM_STATE)**
0	0h RW	**Initiate Force PM State (INIT_FPMS):** When set to '1' force PM state to go to the state indicated in bit 4:1

16.880 Battery Charge (BATTERY_CHARGE_REG)—Offset 80E4h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Enable DM_SRC Battery Charge (EN_DS_BC):** 1 - Always enable battery charge DM_SRC if not connected. Don't wait for portable device detect. (spec. ver. 1.2) 0 - Battery charge spec ver. 1.1.
30:4	0h RO	**RESERVED (RSVD)**
3	0h RW	**Enable Port 3 Battery Charging (EN_P3_BC):** 0 - Battery charging disabled (Physical Port #3) 1 - Battery charging enabled (Physical Port #3)
2	0h RW	**Enable Port 2 Battery Charging (EN_P2_BC):** 0 - Battery charging disabled (Physical Port #2) 1 - Battery charging enabled (Physical Port #2)
1	0h RW	**Enable Port 1 Battery Charging (EN_P1_BC):** 0 - Battery charging disabled (Physical Port #1) 1 - Battery charging enabled (Physical Port #1)
0	0h RW	**Enable Port 0 Battery Charging (EN_P0_BC):** 0 - Battery charging disabled (Physical Port #0) 1 - Battery charging enabled (Physical Port #0)

16.881 Port Watermark (HOST_CTRL_WATERMARK_REG)—Offset 80E8h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 800080h

Bit Range	Default & Access	Field Name (ID): Description
31:16	80h RW	**RBUF water mark (RBUF_WM)**
15:0	80h RW	**XBUF water mark (XBUF_WM)**

16.882 SuperSpeed Port Link Control (HOST_CTRL_PORT_LINK_REG)—Offset 80ECh

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 18010000h

Bit Range	Default & Access	Field Name (ID): Description
31:27	3h RW	**Force LTSSM State (FORCE_LTSSM_ST):** LTSSM state to be forced.
26	0h RW	**Direct Link LTSSM State (DL_LTSSM_ST):** 0: Normal operation mode 1: Direct link to a specific state specified by bit 31:27 . It shall be written 0 in normal operation mode.
25	0h RW	**Direct Link To U0 (DL_U0):** 0: Normal operation mode 1: Direct link to U0. It shall be written 0 in normal operation mode.
24:21	0h RW	**Forced Compliance Pattern (FORCED_CMP_PAT):** Compliance pattern to be forced to enter compliance mode.
20	0h RW	**Enable Link Error Slave Count (EN_LES_CNT):** 0: Disable link error slave count 1: Enable link error slave count
19:17	0h RW	**Debug Mode Select (DEBUG_MD_SEL)**
16:15	2h RW	**PHY Low Power Latency (PHY_LP_LAT):** This field defines the latency to drive the PHY to enter low power mode 0: 4 cycles 1: 8 cycles 2: 16 cycles 3: 32 cycles
14:12	0h RW	**Link Recovery Minimum Time (LR_MIN_TM):** This value defines the minimum time for the link to stay in Recovery.Active other than from U3. The granuity is 128us.
11:9	0h RW	**Link Polling Minimum Time (LP_MIN_TM):** This value defines the minimum time for the link to stay in Polling.Active and Recovery.Active from U3. The granuity is 128us.
8	0h RW	**Force Link Accept PM Command (FORCE_LA_PMC):** 0: Normal operation mode 1: Force link to accept power management command
7	0h RW	**Direct Link Recovery U0 (DL_REC_U0):** 0: Normal operation mode 1: Direct link to Recovery from U0
6	0h RW	**Link Fast Training Mode (LINK_FTM):** 0: Normal operation mode 1: Link fast training mode This bit should be written 0 in normal operation.
5	0h RW	**Disable Link Scrambler (DIS_LINK_SCRAM):** 0: Enable link scrambler 1: Disable link scrambler
4	0h RW	**Direct Link U3 From U0 (DL_U3_U0):** 0: Normal operation mode 1: Direct link to U3 from U0. It shall be written 0 in normal operation mode.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**Direct Link U3 From U0 (DL_U2_U0):** 0: Normal operation mode 1: Direct link to U2 from U0. It shall be written 0 in normal operation mode.
2	0h RW	**Direct Link U3 From U0 (DL_U1_U0):** 0: Normal operation mode 1: Direct link to U1 from U0. It shall be written 0 in normal operation mode.
1	0h RW	**Enable Link Loopback Master Mode (EN_LINK_LB_MAST):** 0: Disable link loopback master mode 1: Enable link loopback master mode
0	0h RW	**Disable Link Compliance Mode (DIS_LINK_CM):** 0: Enable link compliance mode 1: Disable link compliance mode

16.883 USB2 Port Link Control 1 (USB2_LINK_MGR_CTRL_REG1)—Offset 80F0h

These set of registers is used to control jey USB set of timers. They are spread over 4 registers each 32 bits wide.

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 310803A0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	31h RW	**FS/LS Mode SE0 Disconnect Delay[7:0] (FSLS_SE0_DIS_DEL_7_0):** # of microseconds of SE0 in FS/LS mode to register disconnect had occurred.
23:21	0h RW	**Reserved (RSVD0)**
20	0h RW	**L1_EXIT_RECOVERY_MODE (L1_EXIT_RECOVERY_MODE):** Mode for extended L1 Exit recovery delay: 0: 12us 1: 50us
19	1h RW	**L1_TO_INCR_MODE (L1_TO_INCR_MODE):** Mode select for L1 Timeout increments: 0: time out increments are in 125us 1: L1 Timeout increments are in 256us. Refer to USB2 PORTHLPMC.L1 Timeout in XHCI Spec for additional details
18	0h RW	**Reserved (RSVD1)**
17	0h RW	**EN_DETECT_NOMINAL_PKT_EOP (EN_DETECT_NOMINAL_PKT_EOP):** 0: Detect minimal packet EOP. 1: Detect nominal packet EOP.
16	0h RW	**Disable Chirp Response (DIS_CHIRP_RESPONSE):** 0: Normal 1: Force full speed on host ports (disable chirp response)

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**Disable 192 Byte Limit Check (DIS_192B_LIM):** 0: Enforce 192 byte limit on complete-split INs. Treat any packet) 192 as babble case. 1: Disable 192 byte limit check.
14	0h RW	**External Provided FS/LS Disconnect (EXT_FSLS_DIS):** 0: Internal FS/LS Disconnect from linestate(1:0) 1: External provided FS/LS Disconnect from hostdisconnect input
13:12	0h RW	**UTMI Reset Source Select (UTMI_RST_SEL):** Select UTMI Reset Source (FRD UTMI Reset Only) 00: HCReset or Force PHY Reset or internal reset after disconnect/suspend for restart (default) 01,11: UTMI reset = ~UTMI suspendm 10: UTMI reset = ~UTMI suspendm and synchronization to port clk.
11	0h RW	**Disable HS Disconnect Window (DIS_HS_DIS_WIN):** 0: Enable HS Disconnect Window Function 1: Disable HS Disconnect Window Function
10	0h RW	**Disable Port Error Detection (DIS_PERR_DET):** 0: Enable Port Error Detection (default) 1: Disable Port Error Detection
9	1h RW	**Disable Peek Function for ISO-OUT (DIS_PF_IOUT):** 0: Enable Peek function for ISO-OUT (default) 1: Disable Peek function for ISO-OUT
8	1h RW	**Drive Resume-K FS/LS Serial Interface (DRV_RESK_FSLS_SER):** 0: Drive Resume-K on parallel Interface 1: Drive Resume-K directly on FS/LS Serial Interface (default)
7	1h RW	**Enable USB2 Drop-Ping (EN_U2_DROP_PING):** 0: Disable Drop-Ping Function in USB2 Protocol (default) 1: Enable Drop-Ping Function in USB2 Protocol
6	0h RW	**Enable USB2 Force-Ping (EN_U2_FORCE_PING):** 0: Disable Force-Ping Function in USB2 Protocol (default) 1: Enable Force-Ping Function in USB2 Protocol
5	1h RW	**Enable USB2 Auto-Ping (EN_U2_AUTO_PING):** 0: Disable Auto-Ping Function 1: Enable Auto-Ping Function in USB2 Protocol (default)
4	0h RW	**Disable PHY SuspendM (DIS_PHY_SUSM):** 0: PHY is suspend=U3,U2,disconnect (default) 1: Disable PHY SuspendM in All States
3	0h RW	**UTMI Internal Clock Gate Disable (UTMI_INT_CG_DIS):** 0: Normal operation (internal clock gated in U2,U3,disconnect) 1: UTMI Internal Clock Gate Disable
2	0h RW	**Disable PHY SuspendM in Disconnect State (DIS_PSUSM_DS):** 0: PHY is suspendM=0 in Disconnect State (default) 1: Disable PHY SuspendM in Disconnect State
1	0h RW	**Force PHY Reset (FORCE_PHY_RST):** 0: Normal Operation (default) 1: Force PHY Reset
0	0h RW	**USB2 Accelerated Simulation Timing (U2_ACC_SIM_TIM):** 0: Normal Operation (default - FPGA/ASIC) 1: USB2 Accelerated Simulation Timing (default - simulation)

16.884 USB2 Port Link Control 2 (USB2_LINK_MGR_CTRL_REG2)—Offset 80F4h

These set of registers is used to control jey USB set of timers. They are spread over 4 registers each 32 bits wide.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 80C40620h

Bit Range	Default & Access	Field Name (ID): Description
31	1h RW	**Total Reset Duration[0] (TOT_RST_DUR_0):** # of microseconds for total reset duration
30:18	31h RW	**Chirp-K Duration (CHIRPK_DUR):** # of microseconds of Chirp-K to register that a device is chirping
17:5	31h RW	**K/J Disconnect Connect Delay (KJ_DIS_CON_DEL):** # of microseconds of K/J in disconnected state to register connect has occurred.
4:0	0h RW	**FS/LS Mode SE0 Disconnect Delay[12:8] (FSLS_SE0_DIS_DEL_12_8):** # of microseconds of SE0 in FS/LS mode to register disconnect had occurred.

16.885 USB2 Port Link Control 3 (USB2_LINK_MGR_CTRL_REG3)—Offset 80F8h

These set of registers is used to control jey USB set of timers. They are spread over 4 registers each 32 bits wide.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: F865EB6Bh

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RW	**U2 Entry Ignore Linestate Changes Duration[3:0] (U2_IGN_LS_DUR_3_0):** # of microseconds after entering U2, linestate changes are ignored as bus settles
27:15	10CBh RW	**U3 Entry Ignore Linestate Changes Duration (U3_IGN_LS_DUR):** # of microseconds after entering U3, linestate changes are ignored as bus settles
14:0	6B6Bh RW	**Total Reset Duration[15:1] (TOT_RST_DUR_15_1):** # of microseconds for total reset duration

16.886 USB2 Port Link Control 4 (USB2_LINK_MGR_CTRL_REG4)—Offset 80FCh

These set of registers is used to control jey USB set of timers. They are spread over 4 registers each 32 bits wide.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 8003h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**RESERVED (RSVD)**
21:9	40h RW	**U2 Detect Remote Wake Delay (U2D_RWAKE_DEL):** #of microseconds after detecting U2 remote wake condition to reflect K
8:0	3h RW	**U2 Entry Ignore Linestate Changes Duration[12:4] (U2_IGN_LS_DUR_12_4):** # of microseconds after entering U2, linestate changes are ignored as bus settles

16.887 Bandwidth Calc Control (HOST_CTRL_BW_CTRL_REG)—Offset 8100h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 8008h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD)**
15:0	8008h RW	**Reserved (RSVD_1)**

16.888 Host Interface Control (HOST_IF_CTRL_REG)—Offset 8108h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RW	**Rsvd1 (Rsvd1)**
0	1h RW	**Host IF (HOSTIF)**

16.889 Bandwidth Overload Burst (HOST_BW_OV_BURST_REG)—Offset 810Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 8020h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**RESERVED (RSVD)**
23:12	8h RW	**Per Burst Overhead System BW (PB_OVRH_SBW):** BW calculation: Overhead per burst for system BW calculations. see white paper.
11:0	20h RW	**Per Burst Overhead System BW (PB_OVRH_SSBW):** BW calculation: Overhead per burst for SS BW calculations. see white paper.

16.890 USB Max Bandwidth Control 4 (HOST_CTRL_BW_MAX_REG)—Offset 8128h

Access Method

Type: MEM Register
(Size: 64 bits)

Device:
Function:

Default: F42528505647F42h

Bit Range	Default & Access	Field Name (ID): Description
63:60	0h RO	**Reserved (RSVD)**
59:48	F42h RW	**PCIe Max BW Units (PCIE_MAX_BW):** Max. Number of BW units for PCIe (system interface) (denominator in 90% calculation)
47:36	528h RW	**TT Max BW Units (TT_MAX_BW):** Max. Number of BW units for TTs. (denominator in 90% calculation)

Bit Range	Default & Access	Field Name (ID): Description
35:24	505h RW	**FS/LS Max BW Units (FSLS_MAX_BW):** Max. Number of BW units for FS/LS ports. (denominator in 90% calculation)
23:12	647h RW	**HS Max BW Units (HS_MAX_BW):** Max. Number of BW units for HS ports. (denominator in 80% calculation)
11:0	F42h RW	**SS Max BW Units (SS_MAX_BW):** Max. Number of BW units for SS ports. (denominator in 90% calculation)

16.891 USB2 Linestate Debug (LINESTATE_DEBUG_REG)—Offset 8130h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**RESERVED (RSVD)**
27:26	0h RO	**Port 14 UTMI Linestate (P14_UTMI_LS)**
25:24	0h RO	**Port 13 UTMI Linestate (P13_UTMI_LS)**
23:22	0h RO	**Port 12 UTMI Linestate (P12_UTMI_LS)**
21:20	0h RO	**Port 11 UTMI Linestate (P11_UTMI_LS)**
19:18	0h RO	**Port 10 UTMI Linestate (P10_UTMI_LS)**
17:16	0h RO	**Port 9 UTMI Linestate (P9_UTMI_LS)**
15:14	0h RO	**Port 8 UTMI Linestate (P8_UTMI_LS)**
13:12	0h RO	**Port 7 UTMI Linestate (P7_UTMI_LS)**
11:10	0h RO	**Port 6 UTMI Linestate (P6_UTMI_LS)**
9:8	0h RO	**Port 5 UTMI Linestate (P5_UTMI_LS)**
7:6	0h RO	**Port 4 UTMI Linestate (P4_UTMI_LS)**
5:4	0h RO	**Port 3 UTMI Linestate (P3_UTMI_LS)**
3:2	0h RO	**Port 2 UTMI Linestate (P2_UTMI_LS)**

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RO	**Port 1 UTMI Linestate (P1_UTMI_LS)**

16.892 USB2 Protocol Gap Timer (USB2_PROTOCOL_GAP_TIMER_REG)—Offset 8134h

Access Method

Type: MEM Register
(Size: 64 bits)

Device:
Function:

Default: C3C640C05140Ch

Bit Range	Default & Access	Field Name (ID): Description
63:56	0h RO	**Reserved (RSVD)**
55:48	Ch RW	**GAP time after LS TX thru FS hub (LSTX_GAP_TIME)**
47:40	3Ch RW	**GAP time after LS RX thru FS hub (LSRX_GAP_TIME)**
39:32	64h RW	**GAP timer after LS (LS_GAP_TIME)**
31:24	Ch RW	**GAP time after FS (FS_GAP_TIME)**
23:16	5h RW	**GAP time after HS RX (HSRX_GAP_TIME)**
15:8	14h RW	**GAP time after HS TX SOF (HSTXSOF_GAP_TIME)**
7:0	Ch RW	**GAP time HS TX Packet (HSTX_GAP_TIME)**

16.893 USB2 Protocol Bus Timeout Timer (USB2_PROTOCOL_BTO_TIMER_REG)—Offset 813Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 8D4258B8h

Bit Range	Default & Access	Field Name (ID): Description
31:21	46Ah RW	**Bus timeout count for LS (LS_BUS_TO)**
20:10	96h RW	**Bus timeout count for FS (FS_BUS_TO)**

Bit Range	Default & Access	Field Name (ID): Description
9:0	B8h RW	**Bus timeout count for HS (HS_BUS_TO)**

16.894 Power Scheduler Control-0 (PWR_SCHED_CTRL0)—Offset 8140h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: A019132h

Bit Range	Default & Access	Field Name (ID): Description
31:24	Ah RW	**Engine Idle Hysteresis (EIH):** This register controls the min. idle span that has to be observed from the engine idle indicators before the power state flags (xhc_*_idle) will indicate a 1.
23:12	19h RW	**Backbone PLL Shutdown Advance Wake (BPSAW):** This register controls the time before the next scheduled transaction where the Backbone PLL request will assert. Register Format: Bits [11:7] # of 125us uframes Bits [6:0] # of microseconds (0-124)
11:0	132h RW	**Backbone PLL Shutdown Min. Idle Duration (BPSMID):** The sum of this register plus the Backbone PLL Shutdown Advance Wake form to a Total Idle time. When the next scheduled periodic transaction is after present time + Total Idle, the Backbone PLL request will de-assert, allowing the PLL to shutdwon. Register Format: Bits [11:7] # of 125us uframes Bits [6:0] # of microseconds (0-124)

16.895 Power Scheduler Control-2 (PWR_SCHED_CTRL2)—Offset 8144h

These bit enable by EP type those EPs classes that are considered for determining next periodic active interval for pre-wake of the periodic_active signal. EP classes that are disabled may never be observed in setting of the periodic_active signal.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 33Fh

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RO	**Reserved (RSVD)**

Bit Range	Default & Access	Field Name (ID): Description
9	1h RW	**HS Interrupt-OUT Alarm (HS_INT_OUT_ALRM)**
8	1h RW	**HS Interrupt-IN Alarm (HS_INT_IN_ALRM)**
7	0h RW	**SS Interrupt-OUT FC Alarm (SS_INT_OUT_FC_ALRM)**
6	0h RW	**SS Interrupt-IN Alarm (SS_INT_IN_FC_ALRM)**
5	1h RW	**SS Interrupt-OUT & not in FC Alarm (SS_INT_OUT_ALRM)**
4	1h RW	**SS Interrupt-IN & not in FC Alarm (SS_INT_IN_ALRM)**
3	1h RW	**HS ISO-OUT Alarm (HS_ISO_OUT_ALRM)**
2	1h RW	**HS ISO-IN Alarm (HS_ISO_IN_ALRM)**
1	1h RW	**SS ISO-OUT Alarm (SS_ISO_OUT_ALRM)**
0	1h RW	**SS ISO-IN Alarm (SS_ISO_IN_ALRM)**

16.896 AUX Power Management Control (AUX_CTRL_REG2)—Offset 8154h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 81390206h

Bit Range	Default & Access	Field Name (ID): Description
31	1h RW	**DIS_L1P2_EXIT_ON_WAKE_EN (DIS_L1P2_EXIT_ON_WAKE_EN):** This bit disables the dependency on Wake Enables defined in PORTSC for L1P2 exit when in D0
30:28	0h RW	**RESERVED0 (RSVD0)**
27:25	0h RW	**RESERVED (RSVD)**

Bit Range	Default & Access	Field Name (ID): Description
24	1h RW	**Enable L1 exit notification to SNPS PCIe core (EN_L1_EXIT_NOTIF_PCIE):** This bit enables a L1 exit notification to SNPS PCIe core. There is a case where USB ports have waked up and AUX PM module has started the wakeup process. The AUX PM control state got into a wait for P0 state because it needs to wait until PCIe core to signal powerdown state change. Due to the fact that the core is in D3Hot, there is no run_stop bit set such that no internal interrupt will be fired. This causes the LTSSM of PCIe stayed in L1 even though AUX PM has known that it needs an L1 exit. This bit works together with bit21 of this register. 1: enables this feature 0: disables this feature.
23	0h RW	**DISABLE PLC ON DISCONNECT (DIS_PLC_ON_DISCONNECT):** 1: do not assert PLC for disconnection 0: assert PLC for disconnection
22	0h RW	**TREAT_IDLE_AS_TS2_IN_LTSSM_WAIT_4_TS2 (TREAT_IDLE_AS_TS2_IN_LTSSM_WAIT_4_TS2):** This bit enables a feature in PCIe core LTSSM to treat IDLE received as TS2 when LTSSM is in wait for TS2 receive state. This is a function requested from PHY where it is possible to not able to receive TS2 without error. 1: treat Logic IDLE as TS2 received when in some PCIe LTSSM state. 0: disable this feature.
21	1h RW	**Disable p2 overwrite due to the D3HOT where PCIe core enters the L1 (DIS_P2_OVERWRITE_DUE2_D3HOT):** We added a feature where if PCIe core LTSSM enters L1 due to the D3hot, the aux PM control will not start a P2 overwrite function in anticipating for the next L23 enter. 1: disables p2 overwrite due to the D3HOT where PCIe core enters the L1. 0: enables P2 overwrite even if we are in D3Hot.
20	1h RW	**Enable the port to enter U3 automatically when in U1/U2 (ENABLE_AUTO_U3_ENTRY_FROM_U2_U3):** 1: enables the port to enter U3 automatically when in U1/U2 0: disables the port to enter U3 automatically when in U1/U2
19	1h RW	**No linkdown reset is issue during low power state (DIS_LINKDOWN_RST_DURING_LOW_POWER):** No linkdown reset is issue during low power state
18	0h RW	**EN_EXIT_DEEP_SLEEP_IF_PCIE_IN_P0 (EN_EXIT_DEEP_SLEEP_IF_PCIE_IN_P0):** This bit enables a feature in AUX PM module where if PCIe core LTSSM is in P0 for a duration of time, we will exit the deep sleep state. This is for failure control in case. 1: enables this feature 0: disable this feature
17	0h RW	**U2_EXIT_LFPS_TIMER_VALUE (U2_EXIT_LFPS_TIMER_VALUE):** This bit selects U2 exit LFPS timer value 0: 320ns 400ns in 25MHz domain 1: 240ns 320ns in 25MHz domain

Bit Range	Default & Access	Field Name (ID): Description
16	1h RW	**EN_EXIT_DEEP_SLEEP_ON_USB_PORT_WAKEUP (EN_EXIT_DEEP_SLEEP_ON_USB_PORT_WAKEUP):** This bit enables a function that AUX PM module exits from the deep sleep state due to the USB ports wakeup level signal. We have added this feature where USB ports will generated a wakeup level signal to wakeup the AUX PM module if it is in deep sleep state and this level signal will be cleared if the change bits are updated by software. 1: enables this function 0: disables this function which means that a wakeup pulse generated from each USB PortSC event will wake up the AUX PM module from deep sleep if the D3 state is programmed.
15:14	0h RW	**P3_ENTRY_TIMEOUT (P3_ENTRY_TIMEOUT):** This field defines the timeout value to enter P3 mode in U2. 00: 7us 8us 01: 511us 512us 10: disables the timer (0us) 11: disables the timer (0us)
13	0h RW	**Enable U2 P3 Mode (EN_U2_P3):** 0: Disable U2 P3 mode 1: Enable U2 P3 mode
12:11	0h RW	**Fine Debug Mode Select (FINE_DM_SEL)**
10	0h RW	**Enable Low Power State Based Core Clock Gating (EN_LP_CORE_CG):** When set to '1' enable core clock gating based on low power state entered
9	1h RW	**Disable USB3 Port Status Changed Event (DIS_U3_PORT_SCE):** 0: Enable USB3 port status change event generation if any change bit is not cleared 1: Disable USB3 port status change event generation if any change bit is not cleared Bit 12 default 0
8:4	0h RW	**Debug Mode Select Register (DEB_MODE_SEL)**
3	0h RW	**Enable Auto Wakeup Non-IDLE (EN_AWAK_NIDLE):** When set to 1 enables the auto wakeup function when engine has identified non IDLE condition.
2	1h RW	**Enable PM Control P1 Exit P2 (EN_PMC_P1_EXIT_P2):** When set 1 enables the PM control module to transition to P1 instead of P0 when exit P2.
1	1h RW	**Enable PCIe PIPE CLK Isolation (EN_PP_CLK_ISOL):** When set to 1 enables the PCIe PIPE CLK to be isolated when main power is removed.
0	0h RW	**Enable P2 Overwrite P1 Allowed Detect (EN_P2OVRP1_ADET):** When set to 1 enables a function that can detect whether or not enable P2 overwrite P1 function. The condition to get to P2 overwrite is when engine is in idle conditions. This means that there is no ISO EP pending.

16.897 USB2 PHY Power Management Control (USB2_PHY_PMC)—Offset 8164h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: FCh

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD)**
7	1h RW	**EN_CMDM_TXRXB (EN_CMDM_TXRXB):** Enable Command Manager Active indication for Tx/Rx Bias circuit HS Phy PM Policy
6	1h RW	**EN_TTE_TXRXB (EN_TTE_TXRXB):** Enable TTE Active indication for Tx/Rx Bias circuit HS Phy PM Policy
5	1h RW	**EN_IDMA_TXRXB (EN_IDMA_TXRXB):** Enable IDMA Active indication for Tx/Rx Bias circuit HS Phy PM Policy
4	1h RW	**EN_ODMA_TXRXB (EN_ODMA_TXRXB):** Enable ODMA Active indication for Tx/Rx Bias circuit HS Phy PM Policy
3	1h RW	**EN_TRM_TXRXB (EN_TRM_TXRXB):** Enable Transfer Manager Active indication for Tx/Rx Bias circuit HS Phy PM Policy
2	1h RW	**EN_SCH_TXRXB (EN_SCH_TXRXB):** Enable Scheduler Active indication for Tx/Rx Bias circuit HS Phy PM Policy
1	0h RW	**Enable Rx Bias ckt disable (EN_RXB_CD):** When set enables the Rx bias ckt to be disabled when conditions met (as described by the HS phy PM policy bits)
0	0h RW	**Enable Tx Bias ckt disable (EN_TXB_CD):** When set enables the Tx bias ckt to be disabled when conditions met (as described by the HS phy PM policy bits)

16.898 USB Power Gating Control (USB_PGC)—Offset 8168h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 282EEh

Bit Range	Default & Access	Field Name (ID): Description
31:1	14177h RO	**Reserved (RSVD)**
0	0h RW	**USB SRAM power gating enable (USB_SRAM_PGE):** When set enables power gating on USB ports. Usage of this bit is further qualified with xHCI SRAM Dynamic Power Gating Disable fuse. If the fuse disables dynamic power gating, setting this bit to 1 shall not enable power gating feature. This bit always returns the value that was written to it irrespective of the setting of xHCI SRAM Dynamic Power Gating Disable fuse.

16.899 xHCI Aux Clock Control Register (XHCI_AUX_CCR)—Offset 816Ch

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 400h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RO	**Reserved (RSVD)**
19	0h RW	**USB3 Partition Engine/Link trunk gating Enable (PARUSB3_ENG_GEN):** When set to 1 enables gating of the SOSC trunk to the XHCI engine and link in the PARUSB3 partition.
18	0h RW	**USB3 Partition Frame Timer trunk gating Enable (PARUSB3_LINK_GEN):** When set to 1 enables gating of the SOSC trunk to the Frame timer in the PARUSB3 partition.
17	0h RW	**USB2 link partition clock gating enable (PARUSB2_CLK_GEN):** When set to 1 enables gating of the SOSC trunk to the USB2 link and Phy logic in the PARUSB2 partition.
16	0h RW	**USB2/USHIP 12.5 MHz partition clock gating enable (USHIP_PCGEN):** When set to 1 enables gating of the 12.5 MHz SOSC trunk to the USB2 and USHIP logic in the PARUSB2 partition.
15	0h RO	**Reserved1 (RSVD1):** Reserved
14	0h RW	**USB3 Port Aux/Core clock gating enable (USB3_AC_CGE):** When set, allows the aux_cclk clock into the USB3 port to be gated when conditions are met. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.
13:12	0h RW	**Rx Detect Timer when port Aux Clock is Gated (RX_DT_ACG):** This field defines the value of the timer used to perform Rx Detect when port Aux Clock has been gated. 0x0: 100ms 0x1: 12ms Others: Reserved Note: This timer shall use the Fast Training Timer Tick (about 1us tick) for simulation purposes. For Fast Training mode, the above timeouts will become about 11us and about 100us, +/- implementation uncertainty, respectively.
11:8	4h RW	**U2 Residency Before ModPHY Clock Gating (U2R_BM_CG):** Before gating ModPHY Aux clock, Host Controller shall wait for this time in U2. This time is meant to ensure that the attached device has entered U2 as well. 0x0: 1us 0x1: 128us 0x2: 256us 0x3: 512us 0x4: 640us 0x5: 768us 0x6: 896us 0x7: 1024us Others: Reserved Note: This counter shall start counting once pipe has entered PS3 state in response to link in U2.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**Frame Timer Clock Gating Ports in U2 Enable (FTCGPU2E):** This bit, when set, allows Host Controller to gate the clock to the Frame Timer when ports are in U2.
6	0h RW	**USB2 port clock throttle enable (USB2_PC_TE):** When set, allows the Aux clock into the USB2 ports to be throttled when conditions allow.
5	0h RW	**XHCI Engine Aux clock gating enable (XHCI_AC_GE):** When set, allows the aux clock into the XHCI engine to be gated when idle. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.
4	0h RW	**XHCI Aux PM block clock gating enable (XHCI_APMB_CGE):** When set, allows the aux clock into the Aux PM block to be gated when idle. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.
3	0h RW	**USB3 Aux Clock Trunk Gating Enable (USB3_AC_TGE):** When set, allows Aux Clock Trunk feeding to USB3.0 ports to be gated when port Aux clock is gated at all USB3.0 ports and all USB3.0 modPHY instances. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.
2	0h RW	**USB3 Port Aux/Port clock gating enable (USB3_AP_CGE):** When set, allows the aux_pclk clock into the USB3 port to be gated when conditions are met. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.
1	0h RW	**ModPHY port Aux clock gating enable in U2 (MPP_AC_GEU2):** When set, allows the aux clock into the ModPhy to be gated when Link is in U2 and pipe has been in PS3 for at least the time defined by U2 Residency Before ModPHY Clock Gating field. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**ModPHY port Aux clock gating enable in Disconnected, U3 or Disabled (MPP_AC_GE_DDU3):** When set, allows the aux clock into the ModPHY to be gated when Link is in Disconnected, U3 or Disabled state. Usage of this bit is further qualified with xHC Dynamic Clock Gating Disable fuse. If the fuse disables dynamic clock gating, Aux clock gating will not be enabled either. This bit always returns the value that was written to it irrespective of the setting of xHC Dynamic Clock Gating Disable fuse.

16.900 USB LPM Parameters (USB_LPM_PARAM)—Offset 8170h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 96090032h

Bit Range	Default & Access	Field Name (ID): Description
31:22	258h RW	**USB2_LPM_REG_RSM_U3_DET_NORM (USB2_LPM_REG_RSM_U3_DET_NORM)**
21:19	1h RW	**Min U3 Exit LFPS Duration (MIN_U3E_LFPS_D):** Min U3 Exit LFPS Duration This field defines the minimum duration of LFPS driven by Host Controller upon U3 exit LFPS handshake. Note that theres an uncertainty of +-16us in actual duration driven by the Host Controller. 0b000: 96us 0b001: 160us 0b010: 224us 0b011: 288us 0b100: 352us 0b101: 416us 0b110: 480us 0b111: 544us
18:16	1h RW	**Min U2 Exit LFPS Duration (MIN_U2_ELFPS_D):** Min U2 Exit LFPS Duration This field defines the minimum duration of LFPS driven by Host Controller upon U2 exit LFPS handshake. Note that theres an uncertainty of +-16us in actual duration driven by the Host Controller. 0b000: 96us 0b001: 160us 0b010: 224us 0b011: 288us 0b100: 352us 0b101: 416us 0b110: 480us 0b111: 544us
15	0h RW	**Max PING LFPS Rx Detection (XHCI_MAX_PING_LFPS):** This field defines the maximum timing for PING LFPS. If an incoming LFPS will be considered a PING if it has a timing such that it is less than or equal to the selected value. Otherwise it will be considered for the other types of LFPS. 0b Max PING LFPS timing set to 256 ns (32 link clocks) 1b Max PING LFPS timing set to 320 ns (40 link clocks)
14:10	0h RO	**Reserved (RSVD_1)**

Bit Range	Default & Access	Field Name (ID): Description
9:0	32h RW	**xHCI BESL to HIRD Distance (XHCI_BESL_HIRD_DT):** This field defines the gap between BESL and duration of Resume signalling from Host upon Host Initiated Resume from USB2.0 LPM. Default value of this register corresponds to xHCI spec defined 50us value. Value BESL to HIRD Distance 000h 0us 001h 1us 002h 2us 3FFh 1023us

16.901 xHC Latency Tolerance Parameters - LTV Control (XLTP_LTV1)—Offset 8174h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 40047Dh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Disable scheduler direct transition from IDLE to NO requirement (DIS_SDT_IDL_NR):** 0: (default) allow scheduler direct transition from IDLE to NO requirement 1: Disable scheduler direct transition from IDLE to NO requirement
30:26	0h RW	**Reserved (RSVD)**
25	0h RW	**XHCI LTR Transition Policy (XLTRTP):** When '0', LTR messaging state machine transitions from High, Medium, or Low LTR states to Active state upon the Alarm Timer timout and stays in Active until the next service boundary. When '1', the LTR messaging state machine transitions through High ? Med ? Low ? Active states assuming enough latency is available for each transition.
24	0h RW	**XHCI LTR Enable (XLTRE):** This bit must be set to enable LTV messaging from XHCI to the PMC.
23:12	400h RW	**Periodic Active LTV (PA_LTV):** 23:22 Latency Scale 00b : Reserved 01b : Latency Value to be multiplied by 1024 10b : Latency Value to be multiplied by 32,768 11b : Latency Value to be multiplied by 1,048,576 21:12 Latency Value (ns) Defaults to 0 micro seconds
11:0	47Dh RW	**USB2 Port L0 LTV (USB2_PL0_LTV):** 11:10 Latency Scale 00b : Reserved 01b : Latency Value to be multiplied by 1024 10b : Latency Value to be multiplied by 32,768 11b : Latency Value to be multiplied by 1,048,576 9:0 Latency Value (ns) Defaults to 128 Micro Seconds

16.902 xHC Latency Tolerance Parameters LTV Control 2 (XLTP_LTV2)—Offset 8178h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 17FFh

Bit Range	Default & Access	Field Name (ID): Description
31:13	0h RO	**Reserved (RSVD)**
12:0	17FFh RW	**LTV Limit (LTV_LMT):** This register defines a maximum LTR value that is allowed to be advertised to the PMC. This is meant to be used as a workaround or mitigation if issues are discovered with the LTR values generated by the XHC using the defined algorithms. If the LTR value of the XHC is larger than the value in this register field, the value in this field is sent to the PMC instead. Default value is the highest possible - 101b 12:10: Latency Multiplier Field 000b - Value times 1 ns 001b - Value times 32 ns 010b - Value times 1,024 ns 011b - Value times 32,768 ns 100b - Value times 1,048,576 ns 101b - Value times 33,554,432 ns 110b-111b - Not Permitted 9:0: Latency Value Default = 3FFh

16.903 xHC Latency Tolerance Parameters - High Idle Time Control (XLTP_HITC)—Offset 817Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**Reserved (RSVD)**
28:16	0h RW	**Minimum High Idle Time (MHIT):** This is the minimum schedule idle time that must be available before a "High" LTR value can be indicated. This value must be larger than HIWL 12:7 - Time value in # of 125 Micro Seconds Bus Intervals (0 - 8ms) 6:0 - Fractional BI Time value in Micro Seconds (0 - 124 Micro Seconds)
15:13	0h RO	**Reserved (RSVD_1)**
12:0	0h RW	**High Idle Wake Latency (HIWL):** This is the latency to access memory from the High Idle Latency state. This value must be larger than MIWL and LIWL 12:7 - Time value in # of 125 Micro Seconds Bus Intervals (0 - 8ms) 6:0 - Fractional BI Time value in Micro Seconds (0 - 124 Micro Seconds)

16.904 xHC Latency Tolerance Parameters - Medium Idle Time Control (XLTP_MITC)—Offset 8180h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**Reserved (RSVD)**
28:16	0h RW	**Minimum Medium Idle Time (MMIT):** This is the minimum schedule idle time that must be available before a "Medium" LTR value can be indicated. This value must be larger than MIWL 12:7 - Time value in # of 125 Micro Seconds Bus Intervals (0 - 8ms) 6:0 - Fractional BI Time value in Micro Seconds (0 - 124 Micro Seconds)
15:13	0h RO	**Reserved (RSVD_1)**
12:0	0h RW	**Medium Idle Wake Latency (MIWL):** This is the latency to access memory from the Medium Idle Latency state. This value must be larger than LIWL 12:7 - Time value in # of 125 Micro Seconds Bus Intervals (0 - 8ms) 6:0 - Fractional BI Time value in Micro Seconds (0 - 124 Micro Seconds)

16.905 xHC Latency Tolerance Parameters Low Idle Time Control (XLTP_LITC)—Offset 8184h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**Reserved (RSVD)**
28:16	0h RW	**Minimum Low Idle Time (MLIT):** This is the minimum schedule idle time that must be available before a "Low" LTR value can be indicated. This value must be larger than LIWL 12:7 - Time value in # of 125 Micro Seconds Bus Intervals (0 - 8ms) 6:0 - Fractional BI Time value in Micro Seconds (0 - 124 Micro Seconds)
15:13	0h RO	**Reserved (RSVD_1)**

Bit Range	Default & Access	Field Name (ID): Description
12:0	0h RW	**Low Idle Wake Latency (LIWL):** This is the latency to access memory from the Medium Idle Latency state. 12:7 - Time value in # of 125 Micro Seconds Bus Intervals (0 - 8ms) 6:0 - Fractional BI Time value in Micro Seconds (0 - 124 Micro Seconds)

16.906 HOST_CTRL_BW_MAX3_REG (HOST_CTRL_BW_MAX3_REG)—Offset 8188h

Added for CHV

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: F42F42h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Rsvd (Rsvd)**
23:0	F42F42h RW	**MAX_OUT_BW_UNITS_FOR_SS_PORTS (MAX_OUT_BW_UNITS_FOR_SS_PORTS):** Max. Number of OUT BW units for SS ports - denominator in 90% calculation

16.907 THRM_HOST_CTRL_REG (THRM_HOST_CTRL_REG)—Offset 819Ch

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:21	0h RW	**Rsvd (Rsvd)**
20	0h RW	**SSIC Thermal Throttle Ux Mapping (SSIC_THRM_UX_MAPPING):** Controls if U1 or U2 is forced upon the start of thermal throttle OFF period. 0 Force ports into U2 during Thermal Throttle triggered Ux entry. 1 Force ports into U1 during Thermal Throttle triggered Ux entry.

Bit Range	Default & Access	Field Name (ID): Description
19:18	0h RW	**USB3 Thermal Throttle Ux LGO delay (USB3_THRM_UX_LGO_DELAY):** Controls the delay enforced between LMP for FLPMA and LMP for Ux LGO. After sending LPMA ON , wait for pre-defined number of clocks to initiate LGO_U1/ LGO_U2 00: 8 clocks 01: 32 clocks 10: 128 clocks 11: 0 clocks This field does not apply to ports that are not operating in the mode required to issue FLMA ON.
17	0h RW	**THRM_THROTTLING_DISABLE (THRM_THROTTLING_DISABLE):** 0: Thermal throttling is enabled. 1: Thermal throtling is disabled. The host controller ignores the TT control inputs and does not throttle.
16	0h RW	**USB3 Thermal Throttle Ux Mapping (USB3_THRM_UX_MAPPING):** Controls if U1 or U2 is forced upon the start of thermal throttle OFF period. 0 Force ports into U2 during Thermal Throttle triggered Ux entry. 1 Force ports into U1 during Thermal Throttle triggered Ux entry.
15	0h RW	**THROTTLE_PRIORITY_MODE (THROTTLE_PRIORITY_MODE):** 0: Off period has priority: In this case, when the throttle signal is asserted, the host controller enters the off state on the next uframe boundary, and stays in the off state for the prescribed duration or until the end of the 16 uframe throttle period whichever occurs first. On subsequent throttle periods, the off period occurs first and then the on period. 1: The On period has priority: In this case, when the throttle signal is asserted, the host controller completes the required On period first before entering the off period. If the required number of uFrames has already been executed in a 16 uframe throttle window, the controller enters the off period immediately.
14	0h RW	**DISABLE_FORCE_L1_WHEN_THROTTLED (DISABLE_FORCE_L1_WHEN_THROTTLED):** 0: USB2 port will not force L1 entry on throttled ports. L1 entry will be based on the normal idle timeout 1: USB2 ports will attempt to enter L1 immediately after throttled ports are idle.
13	0h RW	**DISABLE_INTERRUPT_THROTTLING (DISABLE_INTERRUPT_THROTTLING):** 0: Interrupt traffic is throttled 1: Interrupt traffic is not throttled
12	0h RW	**DISABLE_ISOCHRONOUS_THROTTLING (DISABLE_ISOCHRONOUS_THROTTLING):** 0: Isochronous traffic is throttled 1: Isochronous traffic is not throttled
11:8	0h RW	**T1_ACTION (T1_ACTION):** # bus intervals to be idle for async traffic out of the 16 interval master period; from 0 to 15.
7:4	0h RW	**T2_ACTION (T2_ACTION):** # bus intervals to be idle for async traffic out of the 16 interval master period; from 0 to 15.
3:0	0h RW	**T3_ACTION (T3_ACTION):** # bus intervals to be idle for async traffic out of the 16 interval master period; from 0 to 15.

16.908 LFPS_PM_CTRL_REG (LFPS_PM_CTRL_REG)—Offset 81A0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Rsvd (Rsvd)**
6:0	0h RW	**LFPS_PM_EN (LFPS_PM_EN)**

16.909 U2PDM (U2PDM)—Offset 81A4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Rsvd (Rsvd)**
8:0	0h RW	**U2PDM (U2PDM)**

16.910 U2PCM (U2PCM)—Offset 81A8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Rsvd (Rsvd)**
8:0	0h RW	**U2PCM (U2PCM)**

16.911 U3PDM (U3PDM)—Offset 81ACh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Rsvd (Rsvd)**
6:0	0h RW	**U3PDM (U3PDM)**

16.912 U3PCM (U3PCM)—Offset 81B0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Rsvd (Rsvd)**
6:0	0h RW	**U3PCM (U3PCM)**

16.913 THRM_HOST_CTRL_REG2 (THRM_HOST_CTRL_REG2)—Offset 81B4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 7Fh

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Rsvd (Rsvd)**
6:0	7Fh RW	**Force LPM Accept Enable (FORCE_LPM_ACCEPT_EN):** Per Port Control to allow for enforcement to be based on device detection if certain devices do not support FLPMA. 0: Do not set FLPMA prior to Ux entry due to TT 1: Set FLPMA prior to Ux entry due to TT

16.914 LFPSONCOUNT_REG (LFPSONCOUNT_REG)—Offset 81B8h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 20C8h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Rsvd (Rsvd)**
17:16	0h RW	**U2P3 LFPS Periodic Sampling Control (XU2P3LPSC):** This field controls the OFF time for the LFPS periodic sampling for SS and SSIC ports in U2P3. If LFPSPM for a port is 1, it will override the OFF time and LFPS receiver will remain OFF permanently. For Fast Sim mode, 500us will be equivalent to 5us. 0x0 Polling Disable. (RXDET Polling will become 100ms.) 0x1 500us OFF Time 0x2 1ms OFF Time 0x3 1.5ms OFF Time
15:10	8h RW	**XLFPSONCNTSSIC (XLFPSONCNTSSIC):** This time would describe the number of clocks SSIC LFPS will remain ON. SSIC LFPS detection operation may be carried out on using RTC clock or Oscillator clock. The value of this register should be adjusted accordingly. For RTC recommended value is 2. For Oscillator clock, recommended value is 8.
9:0	C8h RW	**XLFPSONCNTSS (XLFPSONCNTSS):** This time would describe the number of clocks LFPS will remain ON. LFPS detection operation may be carried out on using RTC clock or Oscillator clock. The value of this register should be adjusted accordingly. For RTC recommended value is 2. For Oscillator clock, recommended value is 200.

16.915 (USB2PMCTRL_REG)—Offset 81C4h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD)**
11	0h RW	**USB2 PHY SUS Power Gate PORTSC Block Policy (U2PSPGPSCBP):** This controls the policy for blocking PORTSC Updates while the USB2 PHY SUS Well is power gated. When set, the controller will block any updates to the PORTSC caused by port status change if the USB2 PHY SUS is power gated. 0 Do not

Bit Range	Default & Access	Field Name (ID): Description
10:8	0h RW	**USB2 PHY SUS Well Power Gate Entry Hysteresis Count (U2PSPGEHC):** This controls the amount of hysteresis time the controller will enforce after detecting the USB2 PHY SUS Power Gate entry condition. 0h 0 clocks 1h 32 clocks 2h 64 clocks 3h 128 clocks 4h 256 clocks 5h 512 clocks 6h 1024 clocks 7h 2048 clocks
7:4	0h RW	**USB2 PHY SUS Power Gate PORTSC Block Policy (U2CLPGLAT):** This field represents the worst case latency for the USB2 Common Lane to enter and exit its power gate state. This fields is required to be compared to a ports HIRD/HIRD value for the ports that have allowed L1 to L2 mapping to determine if the Common Lane can be allowed to power off. If the power gate entry/ exit latency is greater than the HIRD/HIRDD then the common lane should not be allowed to power gate as this will result in a L1 exit violation. 0h 100 us 1h 200 us 2h 300 us Eh 1500us Fh 1600 us
3:2	0h RW	**USB2 PHY SUS Well Power Gate Policy (U2PSUSPGP):** This field controls when to enable the USB2 PHY SUS Well Power Gating when the proper conditions are met. 00 USB2 PHY SUS Power Gating is Disabled. 01 USB2 PHY SUS Power Gating is Enabled in Only D0 (Excludes D0i3 and D3) 10 USB2 PHY SUS Power Gating is Eanabled in only in D0 and D0i3 (Excludes D3) 11 USB2 PHY SUS Power Gating is Eanabled in D0/D0i3/D3
1	0h RW	**USB2 Common Lane Power Gating Enable During L1 to L2 Mapping for USB2 PHY Power Gating (U2CLPGEL1L2):** This field when set enables the controller to allow for the common lane power gating to be enabled when all ports are exposed as in L2 to the USB2 PHY while at least 1 port has been mapped to L2 from L1. This field alone does not guarantee power gating since the L1 HIRD/HIRDD Value must be compared with the PHYs power gate exit latency (U2CLPGLAT) held in this register to ensure that L1 exit is not violated. 0 USB2 Common Lane Power Gating is disabled when any port has been mapped from L1 to L2. 1 USB2 Common Lane Power Gating is allowed when any port has been mapped to L2 from L1 with the additional condition that the HIRD/HIRDD is greater than the PHYs Power Gate exit latency.
0	0h RW	**USB2 Data Lane L1 to L2 Mapping Enable for USB2 PHY Power Gating (U2DLL1L2ME):** This field when set enables the controller to map an L1 entry directly to L2 to allow the USB2 PHY to trigger its Autonomous Power Gating. The USB2 PHY will trigger PG only when in L2 since it does not fully understand the requirements for L1. 0 USB2 L1 to L2 mapping is disabled for all ports 1 USB2 L1 to L2 mapping is enabled for all ports

16.916 ECC_PARITY_ERROR_LOG_REG (ECC_PARITY_ERROR_LOG_REG)—Offset 83F8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C	**COMMAND_PARITY_DETECTED (COMMAND_PARITY_DETECTED):** Command Parity Error detected on received IOSF transaction
30	0h RW/1C	**DATA_PARITY_DETECTED (DATA_PARITY_DETECTED):** Data Parity Error detected on received IOSF transaction
29	0h RW/1C	**ERROR_PRESENT_DETECTED (ERROR_PRESENT_DETECTED):** Error present detected on received IOSF transaction
28:26	0h RO	**RSVD (RSVD):** Reserved
25:21	0h RW/1C	**Correctable ECC Error Source RF (CORRECTABLE_ECC_ERROR_SOURCE_RF):** When set indicates the specific RF the ECC Error was detected on. This is used along with the ECC Source, and Port Info to determine the specific RF on which Correctable ECC Error was seen on. USB2 Port RF: 00000: Reserved 00001: Async. TX Payload 00010: Periodic TX Payload 00011: RX Payload 00100: TTE Periodic TX Payload 00101: TTE RX Payload Others: Reserved USB3 Port RF: 00000: Reserved 00001: Async TX Payload 00010: Periodic TX Payload 00011: RX Payload Others: Reserved XHCI Engine RF: 00000: Reserved 00001: TTE Context 00010: TRM Context 00011: XHCI Completion Collect 00100: Debug Device Completion Collect 00101: DBC-EXI Trace Data
20:15	0h RW/1C	**CORRECTABLE_ECC_ERROR_SOURCE_PORT (CORRECTABLE_ECC_ERROR_SOURCE_PORT):** When set indicates the Port# if the ECC Error was detected on any of the USB2/USB3 Port RFs. This is valid only when bit[1:0] != 00
14:13	0h RW/1C	**CORRECTABLE_ECC_ERROR_SOURCE_LOG (CORRECTABLE_ECC_ERROR_SOURCE_LOG):** When an Correctable ECC is detected for an RF, the corresponding bit is set to '1' 11: Detected Uncorrectable ECC Error on USB3 RFs 10 : Detected Uncorrectable ECC Error on USB2 RFs 01: Detected Uncorrectable ECC Error on XHCI Engine RFs 00 : No Error Note: On detection of first Uncorrectable ECC Error HW shall lock the Correctable log registers (Source, Port, RF) until SW clears the ECC Error log or cleared by reset.

Bit Range	Default & Access	Field Name (ID): Description
12:8	0h RW/1C	**UNCORRECTABLE_ECC_ERROR_SOURCE_RF (UNCORRECTABLE_ECC_ERROR_SOURCE_RF):** When set indicates the specific RF the ECC Error was detected on. This is used along with the ECC Source, and Port Info to determine the specific RF on which uncorrectable ECC Error was seen on. USB2 Port RF: 00000: Reserved 00001: Async. TX Payload 00010: Periodic TX Payload 00011: RX Payload 00100: TTE Periodic TX Payload 00101: TTE RX Payload Others: Reserved USB3 Port RF: 00000: Reserved 00001: Async TX Payload 00010: Periodic TX Payload 00011: RX Payload Others: Reserved XHCI Engine RF: 00000: Reserved 00001: TTE Context 00010: TRM Context 00011: XHCI Completion Collect 00100: Debug Device Completion Collect 00101: DBC-EXI Trace Data
7:2	0h RW/1C	**UNCORRECTABLE_ECC_ERROR_SOURCE_PORT (UNCORRECTABLE_ECC_ERROR_SOURCE_PORT):** When set indicates the Port# if the ECC Error was detected on any of the USB2/USB3 Port RFs. This is valid only when bit[1:0] != 00
1:0	0h RW/1C	**UNCORRECTABLE_ECC_ERROR_SOURCE_LOG (UNCORRECTABLE_ECC_ERROR_SOURCE_LOG):** When an uncorrectable ECC is detected for an RF, the corresponding bit is set to '1' 11: Detected Uncorrectable ECC Error on USB3 RFs 10 : Detected Uncorrectable ECC Error on USB2 RFs 01: Detected Uncorrectable ECC Error on XHCI Engine RFs 00 : No Error Note: On detection of first Uncorrectable ECC Error HW shall lock the Uncorrectable log registers (Source, Port, RF) until SW clears the ECC Error log or cleared by reset.

16.917 ECC_POISONING_CTRL_REG (ECC_POISONING_CTRL_REG)—Offset 83FCh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RW	**ENGINE_RF_ECC_POISONING_VECTOR (ENGINE_RF_ECC_POISONING_VECTOR):** XHCI Engine RFs ECC Poisoning Inversion bits for the 64b or 128bit based on RF width
22:14	0h RW	**USB3_RF_ECC_POISONING_VECTOR (USB3_RF_ECC_POISONING_VECTOR):** USB3 RFs ECC Poisoning Inversion bits for the 64b or 128bit based on RF width

Bit Range	Default & Access	Field Name (ID): Description
13:5	0h RW	**USB2_RF_ECC_POISONING_VECTOR (USB2_RF_ECC_POISONING_VECTOR):** USB2 RFs ECC Poisoning Inversion bits for the 64b or 128bit based on RF width
4:3	0h RO	**RSVD (RSVD)**
2	0h RW	**ENGINE_RF_ECC_POISONING_EN (ENGINE_RF_ECC_POISONING_EN):** Enable ECC Poisoning for XHCI Engine related RFs that support ECC
1	0h RW	**USB3_RF_ECC_POISONING_EN (USB3_RF_ECC_POISONING_EN):** Enable ECC Poisoning for USB3 Port related RFs that support ECC. Setting this bit enables poisoning of USB3 Port related RFs. This applies to all USB3 ports
0	0h RW	**USB2_RF_ECC_POISONING_EN (USB2_RF_ECC_POISONING_EN):** Enable ECC Poisoning for USB2 Port related RFs that support ECC. Setting this bit enables poisoning of USB2 Port related RFs. This applies to all USB2 ports

16.918 USB2_PORT_STATE_REG (USB2_PORT_STATE_REG)—Offset 8400h

Access Method

Type: MEM Register (Size: 64 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:18	0h RO	**RSVD (RSVD)**
17:0	0h RO	**USB2_PORT_STATE_REG (USB2_PORT_STATE_REG):** Per USB2 Port State Register indicating the following states 0x0 - Connected 0x1 - Suspended 0x2 - Disabled 0x3 - Reserved This register is Read Only and will reflect the state of the ports. Status can change at any time and any usage of this information must be done with care and under specific flows where port state change race conditions are properly handled.

16.919 USB3_PORT_STATE_REG (USB3_PORT_STATE_REG)—Offset 8408h

Access Method

Type: MEM Register (Size: 64 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:14	0h RO	**RSVD (RSVD)**
13:0	0h RO	**USB3_PORT_STATE_REG (USB3_PORT_STATE_REG):** Per USB3 Port State Register indicating the following states 0x0 - Connected 0x1 - Suspended 0x2 - Disabled 0x3 - Reserved This register is Read Only and will reflect the state of the ports. Status can change at any time and any usage of this information must be done with care and under specific flows where port state change race conditions are properly handled.

16.920 FUS1_REG (FUS1_REG)—Offset 8410h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 10000h

Bit Range	Default & Access	Field Name (ID): Description
31:21	0h RO	**RSVD (RSVD)**
20:18	0h RO	**DEV_ID (DEV_ID):** Provides a fuse over-ride for the lower 3 bits of device ID.
17	0h RO	**XHC_DPGDIS (XHC_DPGDIS):** When asserted, it indicates that the xHCI will not dynamically power gate the controller.
16	1h RO	**USBR_DIS (USBR_DIS):** When asserted, it indicates that xHCI does not support USBr
15	0h RO	**XHC_FXN_DIS (XHC_FXN_DIS):** When asserted, it indicates the xHCI is fused to function disabled.
14:10	0h RO	**USB2_PORT_COUNT (USB2_PORT_COUNT):** 0x0 = MAX USB2 Ports enabled 0x1 = Max USB2 Ports -1 enabled 0x1F = All ports disabled
9:5	0h RO	**USB3_PORT_COUNT (USB3_PORT_COUNT):** 0x0 = MAX USB3 Ports enabled 0x1 = Max USB3 Ports -1 enabled 0x1F = All ports disabled
4	0h RO	**DEBUG_MODE_ENABLE (DEBUG_MODE_ENABLE):** 0 = USB debug mode disabled 1 = USB debug mode enabled Not used for LPT.
3	0h RO	**XHC_DCGDIS (XHC_DCGDIS):** 0= USB3 (xHC) dynamic clock gating enabled 1= USB3 (xHC) dynamic clock gating disabled
2	0h RO	**REGFILE_DPGDIS (REGFILE_DPGDIS):** 0 = USB3 (xHC) dynamic RF power gating enabled 1 = USB3 (xHC) dynamic RF power gating disabled

Bit Range	Default & Access	Field Name (ID): Description
1	0h RO	**USBIO_PMDIS (USBIO_PMDIS):** 0=USB2 HW LPM and USB3 HW Ux under xHC enabled 1= USB2 HW LPM and USB3 HW Ux under xHC disabled
0	0h RO	**USB2_PLL_SHUTDOWN_DIS (USB2_PLL_SHUTDOWN_DIS):** 0= USB2 PLL Shutdown enabled 1= USB2 PLL Shutdown disabled

16.921 FUS2_REG (FUS2_REG)—Offset 8414h

This register is NOT subject to HW save and restore.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**RSVD (RSVD)**
11:0	0h RO	**PORT_MAP_FUS1 (PORT_MAP_FUS1):** 1:0 Port 1 3:2 Port 2 5:4 Port 3 31:30 Port 16 Bit Description: 00: Port assgned to USB3 01: Port assigned to non-XHCI controller 10: Port assigned to Flex IO mapping. Mapping is based on Soft Straps. 11: Reserved.

16.922 FUS3_REG (FUS3_REG)—Offset 8418h

This register is NOT subject to HW save and restore.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RSVD (RSVD)**

16.923 STRAP1_REG (STRAP1_REG)—Offset 841Ch

This register is NOT subject to HW save and restore.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RO	**RSVD (RSVD)**
5:0	0h RO	**FlexIO mapping (FLEX_IO_MAPPING):** Each bit corresponds to the appropriately numbered USB3 port (zero based) Bit 0 = port 1, bit 1 = port 2 etc. 0: Port is owned by XHCI 1: Port is not owned by XHCI

16.924 STRAP2_REG (STRAP2_REG)—Offset 8420h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RO	**RSVD (RSVD)**
5:0	0h RO	**USB3_SSIC_MODE (USB3_SSIC_MODE):** Each bit corresponds to the appropriately numbered USB3 port (zero based) Bit 0 = port 1, bit 1 = port 2 etc. 0: Port is operated in USB3 mode 1: Port is operated in SSIC mode.

16.925 STRAP3_REG (STRAP3_REG)—Offset 8424h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**RSVD (RSVD)**
1:0	0h RO	**XHCI_PRI_CLK_FREQ_SEL (XHCI_PRI_CLK_FREQ_SEL):** Specifies the frequency of the Primary clock (iosf_prim_mux_clk) used by XHCI 00 : 200 MHz (default) 01 : 125 MHz 10 : 250 MHz

16.926 DFT_REG1 (DFT_REG1)—Offset 8430h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**SS/SSIC DFT CRC (SSDFTCRC):** These register bits contain the value of SS DFT CRC
15:12	0h RO	**RSVD0 (RSVD0):** RSVD0
11	0h RW	**DFTLFPSSEL (DFTLFPSSEL):** DFTLFPSSEL 0: Rxelecidle is driven from GPIO Pin 1: Rxelecidle is Driven as specified by Super Speed DFT LFPS Mode Select
10	0h RW	**Super Speed DFT LFPS Mode Select (SSDFTLMSEL):** This bit selects the Super Speed DFT LFPS mode, and will only take effect when Super Speed HBP mode is enabled. 0b: HBP logic will internally loopback TX LFPS as RX LFPS and AFE RX LFPS path to the controller is disconnected. 1b: RX LFPS path works normally
9:6	0h RW	**SS/SSIC DFT CRC Select (SSDFTCRCSEL):** These bits select which Super Speed DFT CRC value is reflected in SSDFTCRC bits. In addition, these bits also select which SuperSpeed DFT CRC MSB value is sent to GPIO monitor pin, i.e. sata3gp_gp37 to aid silicon debug. Live version of selected CRC's MSB will be- sent to GPIO monitor pin, i.e. sata3gp_gp37. 0h (default): No SuperSpeed DFT CRC is selected. 1h: Data Payload CRC 2h: Link Management Packet CRC 3h: Transaction Packet CRC 4h: Isochronous Timestamp Packet CRC 5h: Data Packet Header CRC 6h: Link Command Packet CRC 7h: RRAP Packet CRC 8h: Tx/Rx Cfg CRC Others: Reserved Note : CRC types 7 and 8 are onlyapplicable if Port is SSIC Port
5	0h RO	**RSVD1 (RSVD1):** RSVD1
4:0	0h RW	**SS/SSIC DFT CRC Port Select (SSICDFTCPS):** One CRC per packet type is shared for all the Super Speed ports. These bits select the Super Speed port for which CRC data will be updated. 000b: (default) No SuperSpeed Port is selected 001b: SS/SSIC Port 0 010b: SS/SSIC Port 1 011b: SS/SSIC Port 2 100b: SS/SSIC Port 3 101b: SS/SSIC PORT4 110b :SS/SSIC Port 5 others : Rsvd

16.927 DFT_REG2 (DFT_REG2)—Offset 8434h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**HS/HSIC TX CRC (TXCRC):** These register bits contain the value of TX CRC. This TX CRC is placed right after the 480MHz XCLKQ.
15:11	0h RW	**HSIC/UTMI+ DFT Port Select (UTMIDFTPS):** One CRC is shared for all the UTMI+ ports. These bits select the UTMI+ port for which CRC data will be updated and loopback status reflected in UTMILPBKSTS. 0h: (default) No UTMI+ Port is selected 1h: UTMI+ Port 0 2h: UTMI+ Port 1 3h: UTMI+ Port 2 4h: UTMI+ Port 3 5h: UTMI+ Port 4 6h: UTMI+ Port 5 7h: UTMI+ Port 6 8h: UTMI+ Port 7 9h: UTMI+ Port 8 Ah: UTMI+ Port 9 Bh: UTMI+ Port 10 Ch: UTMI+ Port 11 Dh: UTMI+ Port 12 Eh: UTMI+ Port 13 Others: Reserved
10:7	0h RW	**Loop Number (UTMILPBKLOOPN_3_0):** Number of repeatable fixed pattern within a packet Note: Connect register bit 3 to counter bit 7, register bit 2 to counter bit 5, register bit 1 to counter bit 3, register bit 0 to counter bit 1. Counter bits 6, 4, 2 and 0 will be tied off to 0. Hence, the programmable loop number shall be: 0000b: 0 loop 0001b: 2 loops 0010b: 8 loops 0011b: 10 loops 1100b: 160 loops 1101b: 162 loops 1110b: 168 loops 1111b: 170 loops (max) +E27
6:5	0h RW	**Operational Mode (UTMIOPMODE_1_0):** Operational Mode in test mode. These signals select between various operational modes: 00b: Normal Operation 01b: Non-Driving 10b: Disable Bit Stuffing and NRZI encoding 11b: Reserved
4	0h RW	**Termination Select (UTMITERMSEL):** Termination Select in test mode. This signal selects between the FS and HS terminations: 0b: HS termination enabled 1b: FS termination enabled
3:2	0h RW	**Transceiver Select (UTMIXCVRSELECT_1_0):** Transceiver Select in test mode. This signal selects between the LS, FS and HS transceivers: 00b: HS transceiver enabled 01b: FS transceiver enabled 10b: LS transceiver enabled 11b: Reserved
1:0	0h RW	**UTMI+ Loopback Status (UTMILPBKSTS):** Loopback Status for port selected by UTMIDFTPS 00b: Reset condition 01b: Comparator has started receiving data and the received data matches with the TX pattern. 10b: Comparator has started receiving data the received data does not match with the TX pattern but there was no assertion of RC error from UTMI. 11b: Comparator has started receiving data and RX ERROR was asserted for at least one clock by UTMI. Note that this does not reflect the status of pattern comparison since RX error from UTMI is unexpected for loopback.

16.928 DFT_REG3 (DFT_REG3)—Offset 8438h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Loopback Pattern (UTMILPBKPAT):** 2-byte pattern for loopback Note: This 2-byte pattern will be replicated to the upper 2-byte to form the DW pattern
15:2	0h RW	**Loopback Lane Select (UTMILPBKSEL_13_0):** Port is selected if the corresponding bit is selected Note: MSB is for UTMI+ Port13, LSB is for UTMI+ Port0
1	0h RW	**Loopback Pattern (UTMILPBKPATSEL):** Near end loopback pattern generation type: 0b: Fixed pattern 1b: USB2 test packet
0	0h RW	**Loopback Type (UTMILPBKTYPE):** Loopback Type in test mode. This signal selects between DNELB and ANELB, and will only take effect if DTUTMILPBKEN is set. 0b: Digital Near-End Loopback (DNELB) 1b: Analog Near-End Loopback (ANELB) Note: Analog Far-End Loopback (AFELB) is not supported

16.929 dft_reg4 (DFT_REG4)—Offset 843Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RW	**RSVD (RSVD)**
10:0	0h RW	**Loopback Pattern (UTMILPBKPATSEL):** Loopback Lane Select (UTMILPBKSEL): Port is selected if the corresponding bit (zero based) is selected Bit 0 = Port 1 Bit 1 = Port 2 Etc.

16.930 dft_reg5 (DFT_REG5)—Offset 8440h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW/L	**High Speed Bypass Enable (HIGH_SPEED_BYPASS_ENABLE):** 0000 - Disable all HBP 0001 - Enable SS HBP 0010 - Enable HS HBP 0011 - Enable HSIC HBP 0100 - Enable SSIC HBP Others - Reserved

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**DFTIDPINSEL (DFTIDPINSEL):** Select Between Device and Host Controller in HBP mode 0h : Device Controller is selected 1h : Host Controller is selected
26	0h RW/O	**Lock Bit (LOCK):** This bit is used by BIOS to lock/unlock lockable bits. When set to '1' the write access to HBP Enable is disabled (locked state). When set to '0', the write access to bit locked are enable (unlocked state). Writable once after platform reset.
25	0h RW/L	**Loopback Enable (UTMILPBKEN):** Enable loopback test mode. If asserted, loopback test mode is enabled
24:0	0h RO	**Rsvd (Rsvd)**

16.931 XECP_CMDM_STS0 (XECP_CMDM_STS0)—Offset 8448h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**IDMA_OWNS_CNTX (IDMA_OWNS_CNTX):** Indicates that IDMA module owns the context access currently
30	0h RO	**ODMA_OWNS_CNTX (ODMA_OWNS_CNTX):** Indicates that ODMA module owns the context access currently
29	0h RO	**TRM_OWNS_CNTX (TRM_OWNS_CNTX):** Indicates that TRM modules owns the context access currently
28	0h RO	**CMD_RING_REQUESTED_CNTX_LOCK (CMD_RING_REQUESTED_CNTX_LOCK):** Indicates that Command Manager has requested a context lock
27	0h RO	**CMD_RING_STOP_IN_PROGRESS (CMD_RING_STOP_IN_PROGRESS):** Indicates that Command Ring stop command is in progress
26	0h RO	**SCH_UPDATE_CLR_EP_IN_PROGRESS (SCH_UPDATE_CLR_EP_IN_PROGRESS):** Indicates that clearing an EP out of schedule is in progress
25	0h RO	**ADDR_DEV_DONE (ADDR_DEV_DONE):** Indicates that current address device command is done by ODMA
24	0h RO	**ADDR_DEV_IN_PROGRESS (ADDR_DEV_IN_PROGRESS):** Indicates that ODMA has an address device command in progress
23	0h RO	**EP_STATE_UPDATE_IN_PROGRESS (EP_STATE_UPDATE_IN_PROGRESS):** Indicates that updating of EP state is in progress

Bit Range	Default & Access	Field Name (ID): Description
22	0h RO	**EP_STATE_IN_PROGRESS_FROM_STOP_DUE_TO_DB (EP_STATE_IN_PROGRESS_FROM_STOP_DUE_TO_DB):** Indicates that doorbell manager is issuing and EP update due to a doorbell ring on an EP that is in stop state
21	0h RO	**EP_STATE_UPDATE_IN_PROGRESS_DUE_TO_EP_ERROR (EP_STATE_UPDATE_IN_PROGRESS_DUE_TO_EP_ERROR):** Indicates that transfer ring manager is isuing and EP update due to an EP error condition detected
20	0h RO	**EP_STATE_UPDATE_IN_PROGRESS_DUE_TO_STALL (EP_STATE_UPDATE_IN_PROGRESS_DUE_TO_STALL):** Indicates that transfere ring manager is issuing an EP state update due to stall received
19	0h RO	**Reserved (RSVD)**
18	0h RO	**STOP in progress (STOP_IN_PROGRESS):** Indicates that a STOP on the Command Ring is in progress
17	0h RO	**commend ring has doorbell pending (CMD_RING_DB_PENDING):** Indicates that the commend ring has doorbell pending
16	0h RO	**commend ring running (CMD_RING_RUNNING):** Indicates that the commend ring is running
15:8	0h RO	**Command next capability offset (CMD_NEXT_CAP_OFFSET)**
7:0	0h RO	**Vendor defined capability ID (VID)**

16.932 XECP_CMDM_STS1 (XECP_CMDM_STS1)—Offset 844Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 3FC0000h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Rsvd0 (Rsvd0)**
25:18	FFh RO	**Event manager Producer Cycle State (EMPCS)**
17:12	0h RO	**Interrupter 7 TRB Count [5:0] (INT7_TRB_CNT)**
11:6	0h RO	**Interrupter 6 TRB Count [5:0] (INT6_TRB_CNT)**
5:0	0h RO	**Interrupter 5 TRB Count [5:0] (INT5_TRB_CNT)**

16.933 XECP_CMDM_STS2 (XECP_CMDM_STS2)—Offset 8450h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RO	**Rsvd0 (Rsvd0)**
4:0	0h RO	**Event Ring Segment Table (ERST):** count low

16.934 XECP_CMDM_STS3 (XECP_CMDM_STS3)—Offset 8454h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd0 (Rsvd0)**
15:0	0h RO	**Event Ring Segment Table (ERST):** count high

16.935 XECP_CMDM_STS4 (XECP_CMDM_STS4)—Offset 8458h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Event Ring Enqueue Pointer Low (EREPL)**

16.936 XECP_CMDM_STS5 (XECP_CMDM_STS5)—Offset 845Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Event Ring Enqueue Pointer High (EREPH)**

16.937 AUX Power PHY Reset (UPORTS_PON_RST_REG)—Offset 8460h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**RESERVED (RSVD)**
3:0	0h WO	**Allow Software USB PHY RST (ALL_SW_UP_RST):** Allow a USB PHY reset being issued by software. Writing to this register with bit set to 1 will reset the USB PHY that is connected to the port. Bit3:0 indicates the port number of the USB PHY

16.938 Latency Tolerance Control 0 (HOST_IF_LAT_TOL_CTRL_REG0)—Offset 8464h

The Latency Tolerance Control Register is used by SW to control which BELT is returned when this register is read. SW shall write to this register to program a Slot-ID, Port-ID and BELT Select to determine which BELT is selected. When this register is read the selected BELT is returned.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: D0000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h WO	**BELT Select (BELT_SEL):** This field determines what value will be selected to read back from SW when reading this register 0: Returns the SW programmed Latency Tolerance Value 1: Returns the Lowest BELT in the Host 2: Returns the BELT for the requested Slot-ID (Slot Select) 3: Returns the BELT for the requested Port-ID (Port Select)
29:20	0h RO	**Rsvd1 (Rsvd1)**
19:16	Dh WO	**Port Select (PORT_SEL):** Used to select the BELT for a given Port # when the BELT Select is programmed to select the Port-ID (this field is 0 based)
15:12	0h RO	**Rsvd (Rsvd)**
11:5	0h RO	**BELT Value (BELTV):** Value of selected BELT is return in this field
4:0	0h RW	**Slot Select (SLOT_SEL):** Reads will return: BELT Value (BELTV) [4:0]: Value of selected BELT is return in this field Writes will control : Slot Select (): Used to select the BELT for a given Slot # when the BELT Select is programmed to select the Slot-ID (this field is zero based)

16.939 USB Legacy Support Capability (USBLEGSUP)—Offset 846Ch

This register is modified and maintained by BIOS

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 2201h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Rsvd2 (Rsvd2)**
24	0h RW	**HC OS Owned Semaphore (HCOSOS)**
23:17	0h RO	**Rsvd1 (Rsvd1)**
16	0h RW	**HC BIOS Owned Semaphore (HCBIOSOS)**
15:8	22h RW/S	**Next Capability Pointer (NextCP)**
7:0	1h RW/L	**Capability ID (CID)**

16.940 USB Legacy Support Control Status (USBLEGCTLSTS)—Offset 8470h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/C	**SMI on BAR (SMIBAR)**
30	0h RW/C	**SMI on PCI Command (SMIPCIC)**
29	0h RW/C	**SMI on OS Ownership Change (SMIOSOC)**
28:21	0h RO	**Rsvd4 (Rsvd4)**
20	0h RO	**SMI on Host System Error (SMIHSE)**
19:17	0h RO	**Rsvd3 (Rsvd3)**
16	0h RO	**SMI on Event Interrupt (SMIEI)**
15	0h RW	**SMI on BAR Enable (SMIBARE)**
14	0h RW	**SMI on PCI Command Enable (SMIPCICE)**
13	0h RW	**SMI on OS Ownership Enable (SMIOSOE)**
12:5	0h RO	**Rsvd2 (Rsvd2)**
4	0h RW	**SMI on Host System Error Enable (SMIHSEE)**
3:1	0h RO	**Rsvd1 (Rsvd1)**
0	0h RW	**USB SMI Enable (USBSMIE)**

16.941 Port Disable Override capability register (PDO_CAPABILITY)—Offset 84F4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 3C6h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd (Rsvd)**
15:8	3h RO	**Next Capability Pointer (NCP)**
7:0	C6h RO	**Capability ID (CID)**

16.942 USB2 Port Disable Override (USB2PDO)—Offset 84F8h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Rsvd1 (Rsvd1)**
8:0	0h RW/O	**USB2PDO (USB2PDO)**

16.943 USB3 Port Disable Override (USB3PDO)—Offset 84FCh

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Rsvd1 (Rsvd1)**
6:0	0h RW/O	**USB3 Port Disable Override (USB3PDO):** A '1' in a bit position prevents the corresponding USB3 port from reporting a Device Connection to the XHC.

16.944 HW state capability register (HW_STATE_CAPABILITY)—Offset 8500h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 1F40C7h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Rsvd (Rsvd)**
23:16	1Fh RO	**Save Length (SAVE_LENGTH):** Indicates the number of DWords in this capability starting at offset 04h, that need to be saved and restored
15:8	40h RO	**Next Capability Pointer (NCP)**
7:0	C7h RO	**Capability ID (CID)**

16.945 HW state register 1 (HW_STATE_REG1)—Offset 8504h

Implementation defined HW state: SW must not access this register.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**HW_STATE_REG1 (HW_STATE_REG1)**

16.946 HW state register 2 (HW_STATE_REG2)—Offset 8508h

Implementation defined HW state: SW must not access this register.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**HW_STATE_REG2 (HW_STATE_REG2)**

16.947 HW state register 3 (HW_STATE_REG3)—Offset 850Ch

Implementation defined HW state: SW must not access this register.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**HW_STATE_REG3 (HW_STATE_REG3)**

16.948 HW state register 4 (HW_STATE_REG4)—Offset 8510h

Implementation defined HW state: SW must not access this register.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**HW_STATE_REG4 (HW_STATE_REG4)**

16.949 CONFIG mirror capability register (CONFIG_MIRROR_CAPABILITY)—Offset 8600h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 40C2h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd (Rsvd)**
15:8	40h RO	**Next Capability Pointer (NCP)**
7:0	C2h RO	**Capability ID (CID)**

16.950 Command (CMD_MMIO)—Offset 8604h

Dummy register, mirror of physical register as CMD

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:11	0h RO	**Reserved (RSVD)**
10	0h RW	**Interrupt Disable (ID):** When cleared to 0, the function is capable of generating interrupts. When 1, the function can not generate its interrupt to the interrupt controller. Note that the corresponding Interrupt Status bit is not affected by the interrupt enable.
9	0h RO	**Fast Back to Back Enable (FBE)**
8	0h RW	**SERR# Enable (SERR):** When set to 1, the XHC is capable of generating (internally) SERR#.
7	0h RO	**Wait Cycle Control (WCC)**
6	0h RW	**Parity Error Response (PER):** When set to 1, the XHCI Host Controller will check for correct parity (on its internal interface) and halt operation when bad parity is detected during the data phase as recommended by the XHCI specification. Note that this applies to both requests and completions from the system interface. This bit must be set in order for the parity errors to generate SERR#.
5	0h RO	**VGA Palette Snoop (VPS)**
4	0h RO	**Memory Write Invalidate (MWI)**
3	0h RO	**Special Cycle Enable (SCE)**
2	0h RW	**Bus Master Enable (BME):** When set, it allows XHC to act as a bus master. When cleared, it disable XHC from initiating transactions on the system bus.
1	0h RW	**Memory Space Enable (MSE):** This bit controls access to the XHC Memory Space registers. If this bit is set, accesses to the XHC registers are enabled. The Base Address register for the XHC should be programmed before this bit is set.
0	0h RO	**I/O Space Enable (IOSE):** Reserved as 0. Read-Only.

16.951 Device Status (STS_MMIO)—Offset 8606h

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 290h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**Detected Parity Error (DPE):** This bit is set by the Intel PCH whenever a parity error is seen on the internal interface to the XHC host controller, regardless of the setting of bit 6 or bit 8 in the Command register or any other conditions. Software clears this bit by writing a 1 to this bit location.
14	0h RW/1C	**Signaled System Error (SSE):** This bit is set by the Intel PCH whenever it signals SERR# (internally). The SERR_EN bit (bit 8 in the Command Register) must be 1 for this bit to be set. Software clears this bit by writing a 1 to this bit location.
13	0h RW/1C	**Received Master-Abort Status (RMA):** This bit is set when XHC, as a master, receives a master-abort status on a memory access. This is treated as a Host Error and halts the DMA engines. Software clears this bit by writing a 1 to this bit location.
12	0h RW/1C	**Received Target Abort Status (RTA):** This bit is set when XHC, as a master, receives a target abort status on a memory access. This is treated as a Host Error and halts the DMA engines. Software clears this bit by writing a 1 to this bit location.
11	0h RW/1C	**Signaled Target-Abort Status (STA):** This bit is used to indicate when the XHC function responds to a cycle with a target abort.
10:9	1h RO	**DEVSEL# Timing Status (DEVT):** This 2-bit field defines the timing for DEVSEL# assertion. Read-Only.
8	0h RW/1C	**Master Data Parity Error Detected (MDPED):** This bit is set by the Intel PCH whenever a data parity error is detected on a XHC read completion packet on the internal interface to the XHC host controller and bit 6 of the Command register is set to 1. Software clears this bit by writing a 1 to this bit location.
7	1h RO	**Fast Back-to-Back Capable (FBBC):** Reserved as 1 Read-Only.
6	0h RO	**User Definable Features (UDF):** Reserved as 0. Read-Only.
5	0h RO	**66 MHz Capable (MC):** Reserved as 0. Read-Only.
4	1h RO	**Capabilities List (CL):** Hardwired to 1 indicating that offset 34h contains a valid capabilities pointer.
3	0h RO/V	**Interrupt Status (IS):** This read-only bit reflects the state of this function's interrupt at the input of the enable/disable logic. This bit is a 1 when the interrupt is asserted. This bit will be 0 when the interrupt is deasserted. The value reported in this bit is independent of the value in the Interrupt Enable bit.
2:0	0h RO	**Reserved (RSVD)**

16.952 Revision ID (RID_MMIO)—Offset 8608h

Access Method

Type: MEM Register
(Size: 8 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO/V	**Revision ID (RID):** See Chap 6 for value.

16.953 Programming Interface (PI_MMIO)—Offset 8609h

Access Method

Type: MEM Register
(Size: 8 bits)

Device:
Function:

Default: 30h

Bit Range	Default & Access	Field Name (ID): Description
7:0	30h RO	**Programming Interface (PI):** A value of 30h indicates that this USB Host Controller conforms to the XHCI specification.

16.954 Sub Class Code (SCC_MMIO)—Offset 860Ah

Access Method

Type: MEM Register
(Size: 8 bits)

Device:
Function:

Default: 3h

Bit Range	Default & Access	Field Name (ID): Description
7:0	3h RO	**Sub Class Code (SCC):** A value of 03h indicates that this is a Universal Serial Bus Host Controller.

16.955 Base Class Code (BCC_MMIO)—Offset 860Bh

Access Method

Type: MEM Register
(Size: 8 bits)

Device:
Function:

Default: Ch

Bit Range	Default & Access	Field Name (ID): Description
7:0	Ch RO	**Base Class Code (BCC):** A value of 0Ch indicates that this is a Serial Bus controller.

16.956 Master Latency Timer (MLT_MMIO)—Offset 860Dh

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Master Latency Timer (MLT):** Because the XHC controller is internally implemented with arbitration on an internal interface, it does not need a master latency timer. The bits will be fixed at 0.

16.957 Header Type (HT_MMIO)—Offset 860Eh

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 80h

Bit Range	Default & Access	Field Name (ID): Description
7	1h RO	**Multi-Function Bit (MFB):** Read only indicating single function device.
6:0	0h RO	**Configuration layout (CL):** Hardwired to 0 to indicate a standard PCI configuration layout.

16.958 Memory Base Address (MBAR_MMIO)—Offset 8610h

Dummy register, mirror of physical register as MBAR. Value in this register will be different after the enumeration process.

Access Method

Type: MEM Register (Size: 64 bits)
Device:
Function:

Default: 4h

Bit Range	Default & Access	Field Name (ID): Description
63:16	0h RW	**Base Address (BA):** Bits (63:16) correspond to memory address signals (63:16), respectively. This gives 64 KB of relocatable memory space aligned to 64 KB boundaries.
15:4	0h RO	**Reserved (RSVD):** Reserved. Read-Only 0, this indicates that this function is requesting an 64KB block of memory.
3	0h RO	**Prefetchable (Prefetchable):** This bit is hardwired to 0 indicating that this range should not be prefetched.
2:1	2h RO	**Type (Type):** If this field is hardwired to 00 it indicates that this range can be mapped anywhere within 32-bit address space. If this field is hardwired to 10 it indicates that this range can be mapped anywhere within 64-bit address space.
0	0h RO	**Resource Type Indicator (RTE):** This bit is hardwired to 0 indicating that the base address field in this register maps to memory space

16.959 USB Subsystem Vendor ID (SSVID_MMIO)—Offset 862Ch

Dummy register, mirror of physical register as SSVID. This register is modified and maintained by BIOS

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW/L	**USB Subsystem Vendor ID (SSVID):** This register, in combination with the USB Subsystem ID register, enables the operating system to distinguish each subsystem from the others.

16.960 USB Subsystem ID (SSID_MMIO)—Offset 862Eh

Dummy register, mirror of physical register as SSID. This register is modified and maintained by BIOS

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW/L	**USB Subsystem ID (SSID):** BIOS sets the value in this register to identify the Subsystem ID. This register, in combination with the Subsystem Vendor ID register, enables the operating system to distinguish each subsystem from other(s).

16.961 Capabilities Pointer (CAP_PTR_MMIO)—Offset 8634h

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 70h

Bit Range	Default & Access	Field Name (ID): Description
7:0	70h RO	**Capabilities Pointer (CAP_PTR):** This register points to the starting offset of the capabilities ranges.

16.962 Interrupt Line (ILINE_MMIO)—Offset 863Ch

Dummy register, mirror of physical register as ILINE.

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Interrupt Line (ILINE):** This data is not used by the Intel PCH. It is used as a scratchpad register to communicate to software the interrupt line that the interrupt pin is connected to.

16.963 Interrupt Pin (IPIN_MMIO)—Offset 863Dh

Dummy register, mirror of physical register as IPIN.

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW/L	**Interrupt pin (IPIN):** Bits 7:0 reflect the Interrupt Pin assigned to the host controller by the platform (and are hardwired).

16.964 XHC System Bus Configuration 1 (XHCC1_MMIO)—Offset 8640h

Dummy register, mirror of physical register as XHCC1

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 1FDh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/O	**Access Control (ACCTRL):** This bit is used by BIOS to lock/unlock lockable bits. When set to '1' the write access to bits locked by this bit is disabled (locked state). When set to '0', the write access to bit locked by this bit is enabled (unlocked state). Writable once after platform reset.
30:25	0h RW	**ECO1 (ECO1):** Reserved
24	0h RW	**Master/Target Abort SERR (RMTASERR):** When set, it allows the out-of-band error reporting from the xHCI Controller to be reported as SERR# (if SERR# reporting is enabled) and thus set the STS.SSE bit.
23	0h RW/C	**Unsupported Request Detected (URD):** Set the HW when xHCI Controller received an unsupported request posted cycle. Cleared by SW when the bit is written with value of '1'.
22	0h RW	**Unsupported Request Report Enable (URRE):** When set this bit allows the URD bit to be reported as SERR# (if SERR# reporting is enabled) and thus set the STS.SSE bit.
21:19	0h RW	**Inactivity Initiated L1 Enable (IIL1E):** If programmed to non-zero, it allows L1 power managed to be enabled after the time-out period specified. 000: Disabled 001: 32 bb_cclk 010: 64 bb_cclk 011: 128 bb_cclk 100: 256 bb_cclk 101: 512 bb_cclk 110: 1024 bb_cclk 111: 131072 bb_cclk
18	0h RW	**XHC Initiated L1 Enable (XHCIL1E):** If set, allow the XHC initiated L1 power mangement to be enabled.
17	0h RW	**D3 Initiated L1 Enable (D3IL1E):** If set, allow PCI device state D3 initiated L1 power managment to be enables.

Bit Range	Default & Access	Field Name (ID): Description
16:12	0h RW	**Periodic Complete Pre Wake Time (PCPWT):** The value programmed in this field determines how far in advance of the start of the next micro-frame the host controller must de-assert the "Periodic Complete" signal . This allows for platform wake time before the next scheduled periodic transaction. The value programmed in this field represents the # of bytes consumed in the current micro-frame which is required to allow for the periodic complete to de-assert. This allows for a programmable time to cause the periodic complete to de-assert prior to the start of the next micro-frame. Register Format: Bits (16:12) represnets the # of bytes remaining with a 256B granularity. Periodic Complete will de-assert if the bytes consumed in the current micro-frame is less
11	0h RW	**SW Assisted xHC Idle (SWAXHCI):** This bit being set will indicate xHC idleness (through SW means), which must be a '1' to allow L1 entry, and subsequently allow backbone clock to be gated. This bit is to be set by Intel xHCI driver after checking that the xHCI Controller will stay in idle state for a significant period of time, e.g. all ports disconnected. This bit can be cleared under the following conditions (see SWAXHCI Policy bits in xHC System Bus Configuration 2 register): n SW: SW could write 0 to clear this bit. n HW: HW, under policy control, will clear this bit on an MMIO access to the Host Controller. n HW: HW, under policy control, will clear this bit when HW exits Idle state.
10:8	1h RW	**L23 to Host Reset Acknowledge Wait Count (L23HRAWC):** If programmed to non zero, it allows a wait period after the L23 PHY has shutdown before returning host reset acknowledge to PMC. 000: Disabled 001: 128 bb_cclk 010: 256 bb_cclk 011: 512 bb_cclk 100: 1024 bb_cclk 101: 2048 bb_cclk 110: 4096 bb_cclk 111: 131072 bb_cclk
7:6	3h RW	**Upstream Type Arbiter Grant Count Posted (UTAGCP):** Grant count for IOSF upstream L2 request type arbiter for posted type
5:4	3h RW	**Upstream Type Arbiter Grant Count Non Posted (UDAGCNP):** Grant count for IOSF upstream L2 type arbiter for non-posted type
3:2	3h RW	**Upstream Type Arbiter Grant Count Completion (UDAGCCP) (UDAGCCP):** Grant count for IOSF upstream L2 type arbiter for completion type
1:0	1h RW	**Upstream Device Arbiter Grant Count (UDAGC) (UDAGC):** Grant count for IOSF upstream L1 device arbiter

16.965 XHC System Bus Configuration 2 (XHCC2_MMIO)—Offset 8644h

Dummy register, mirror of physical register as XHCC2

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 3C000h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**OC Configuration Done (OCCFGDONE):** This bit is used by BIOS to prevent spurious switching during OC configuration. It must be set by BIOS after configuration of the OC mapping bits is complete. Once this bit is set, OC mapping shall not be changed by SW.
30:26	0h RW	**ECO1 (ECO1):** Reserved
25	0h RW	**DMA Request Boundary Crossing Control (DREQBCC):** This bit controls the boundary crossing limit of each Read/Write Request. 0: 4KB 1: 64B
24:22	0h RW	**IDMA Read Request Size Control (IDMA_RDREQSZCTRL):** Read Request Size Control: This bit controls the maximum size of each Read Request. 000: 128B 001: 256B 011 - 110: Reserved 111: 64B
21	0h RW	**XHC Upstream Read Relaxed Ordering Enable (XHCUPRDROE):** This policy controls the Relaxed Ordering attribute for upstream reads. 0 - xHC will clear RO for all upstream read requests. 1 - xHC will set RO for all upstream read requests.
20	0h RW	**IOSF Sideband Register Access Disable (IOSFSRAD):** When set, it disables the IOSF sideband interface from accepting any host space register access.
19:14	Fh RW	**Upstream Non-Posted Pre-Allocation (UNPPA):** This field reserves data sizes, in 64 byte chunks, of the downstream completion resource. This value is zero based. 000000 - 111111: Pre-allocate 64 bytes - 4096 bytes If set greater than the default allows over-allocation If set less than default allows under-allocation Only allowed to be programmed when BME = 0 and no outstanding downstream completion
13:12	0h RW	**SW Assisted xHC Idle Policy (SWAXHCIP):** Note: Irrespective of the setting of this field, SW write of 0 to SWAXHCI will clear the bit. 00b (default): xHC HW clears SWAXHCI bit upon: n MMIO access to Host Controller OR n xHC HW exits Idle state 01b: xHC HW does not autonomously clear SWAXHCI bit. The bit could be cleared only by SW. 10b: xHC HW clears SWAXHCI upon MMIO acces to Host Controller. xHC HW exit from Idle state will not clear SWAXHCI. 11b: Reserved
11	0h RW	**MMIO Read After MMIO Write Delay Disable (RAWDD):** This field controls delay on MMIO Read after MMIO Write. 0b (Default): Delay MMIO Read after MMIO Write 1b: Do not delay MMIO Read after MMIO Write Note that this delay applies after the second of the two DW writes in the case where the IOSF Gasket splits a QW write into two single DW writes to the IP.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW	**MMIO Write After MMIO Write Delay Enable (WAWDE):** This field controls delay on MMIO Write after previous MMIO Write. 0b (Default): Do not delay MMIO Write after previous MMIO Write 1b: Delay MMIO Write after previous MMIO Write Note that the delay count does not apply on the second of the two DW writes that are generated by IOSF Gasket when it splits a QW write into two. In other words, the second of the two DW writes could happen without any delay with respect to the first DW write. This choice is being made for ease of ECO. The delay count, in this case, will apply after the second of the two DW writes.
9:8	0h RW	**SW Assisted Cx Inhibit (SWACXIHB):** This field controls how the DMI L1 inhibit signal from USB3 to PMC will behave. 00: Never inhibit Cx 01: Inhibit Cx when Isochronous Endpoint is active (PPT Behavior) 10: Inhibit Cx when Periodic Active as defined in 40.4.3.2.1 11: Always inhibit Cx
7:6	0h RW	**SW Assisted DMI L1 Inhibit (SWADMIL1IHB):** This field controls how the DMI L1 inhibit signal from USB3 to DMI will behave. 00: Never inhibit DMI L1. 01: Inhibit DMI L1 when Isochronous Endpoint is active (PPT Behavior). 10: Inhibit DMI L1 when Priodic Active as defined in 40.4.3.2.1. 11: Inhibit DMI L1 if XHCC1.SWAXHCI = 0.
5:3	0h RW	**L1 Force P2 Clock Gating Wait Count (L1FP2CGWC):** If programmed to non zero, it allows L1 force P2 gating off the clock to be delayed after the time-out period specified. If wake up event is detected before the time-out, pclk remains alive and trigger L1 exit as though CPU host is causing the wake, 000: Disabled 001: 128 bb_cclk 010: 256 bb_cclk 011: 512 bb_cclk 100: 1024 bb_cclk 101: 2048 bb_cclk 110: 4096 bb_cclk 111: 131072 bb_cclk
2:0	0h RW	**Read Request Size Control (RDREQSZCTRL):** Read Request Size Control: This bit controls the maximum size of each Read Request. 000: 128B 001: 256B 010: 512B 011 - 110: Reserved 111: 64B

16.966 Clock Gating (XHCLKGTEN_MMIO)—Offset 8650h

Dummy register, mirror of physical register as XHCLKGTEN

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**Rsvd2 (Rsvd2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
28	0h RW	**Nak'ing USB2.0 EPs for Backbone Clock Gating and PLL Shutdown (NUEFBCGPS):** This field controls whether Naking USB2.0 EPs, once in Naking low priority schedule, should be considered as active for the considerations for backbone clock gating and PLL shutdown or not. 0: Naking USB2.0 EPs are not considered to be active for Backbone Clock Gating and PLL Shutdown evaluation. 1: Naking USB2.0 EPs are considered to be active for Backbone clock gating and PLL shutdown evaluation.
27	0h RW	**SRAM Power Gate Enable (SRAMPGTEN):** This register enables the SRAM Power Gating when PLL shutdown conditions for all clock domains have been met 0 - Disallow SRAM Power Gating. 1 - Allow SRAM Power Gating
26	0h RW	**SS Link PLL Shutdown Enable (SSLSE):** This register enables the SS P3 state to be exposed to PXP PLL Shutdown conditions on behalf of all USB SS Ports ontop of trunk clock gating. 0 - P3 state NOT allowed to result in PXP PLL shutdown. 1- P3 state allowed to result in PXP PLL shutdown
25	0h RW	**USB2 PLL Shutdown Enable (USB2PLLSE):** When set, this bit allows USB2 PLL to be shutdown when HS Link trunk clock is gated, and xHC can tolerate PLL spin up time for subsequent clock request. Note: if USB2 PLL Shutdown Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value.
24	0h RW	**IOSF Sideband Trunk Clock Gating Enable (IOSFSTCGE):** When set, this bit allows the IOSF sideband clock trunk to be gated when idle conditions are met.
23:20	0h RW	**HS Backbone PXP Trunk Clock Gate Enable (HSTCGE):** This register determines the HS Ux state(s) which will be exposed to Backbone PXP trunk gating of core clock. Uy is a state allowed to result in trunk gating when ss_tcg_ux_en(x) is asserted. (0) ==) U0 or deeper (1) ==) NA (no support for U1) (2) ==) U2 (L1) or deeper (3) ==) U3 (L2) or deeper
19:16	0h RW	**SS Backbone PXP Trunk Clock Gate Enable (SSTCGE):** This register determines the SS Ux state(s) which will be exposed to Backbone PXP trunk gating of core clock. Uy is a state allowed to result in trunk gating when ss_tcg_ux_en(x) is asserted. (0) ==) U0 or deeper (1) ==) U1 or deeper (2) ==) U2 or deeper (3) ==) U3 or deeper
15	0h RW	**XHC Ignore_EU3S (XHCIGEU3S):** This register determines if the xHC will use the EU3S as a condition to allow for Frame timer gating. 0 - xHC may allow frame timer to be gated when EU3S is set and all ports are in the required state. 1 - xHC may allow frame timer to be gated regardless of EU3S.
14	0h RW	**XHC Frame Timer Clock Shutdown Enable (XHCFTCLKSE):** This register determines if the xHC will allow the frame timer clock to be gated. 0 - xHC will not allow ICC PLL 96MHz output to be shutdown thus keeping the frame timer running. 1 - xHC will allow ICC PLL 96MHz output to be shutdown under specific conditions.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**XHC Backbone PXP Trunk Clock Gate In Presence of ISOCH EP (XHCBBTCGIPISO):** This register controls the policy on allowing Backbone PXP trunk clock gate in the presence of IDLE ISOCH EP s with active DB. Allows the periodic active to be used in enabling Backbone PXP trunk clock gating of core clock. 0 Trunk gate of core clock is not allowed when ISOCH EP DB is set and ISOCH EP s are idle. 1 Allow trunk gate of core clock when ISOCH EP DB is set and ISOCH EP s are idle.
12	0h RW	**XHC HS Backbone PXP Trunk Clock Gate U2 non RWE (XHCHSTCGU2NRWE):** This register controls the policy on allowing Backbone PXP trunk gating of core clock when there is atleast 1 non Remote Wake Enabled HS Port in U2. 0 Prevent trunk gate of core clock when a non RWE HS Port is in U2. 1 Allow trunk gate of core clock when a non RWE HS Port is in U2.
11:10	0h RW	**XHC USB2 PLL Shutdown Lx Enable (XHCUSB2PLLSDLE):** This register determines the HS Link state(s) which will be exposed to USB2 PLL Shutdown conditions on behalf of all USB2 HS Ports. Ly is a state allowed to result in USB2 PLL Shutdown when en(x) is asserted. (0) ==) L1 or deeper (1) ==) L2 or deeper
9:8	0h RW	**HS Backbone PXP PLL Shutdown Ux Enable (HSUXDMIPLLSE):** This register determines the Ux state(s) which will be exposed to PXP PLL Shutdown conditions. PLL Shutdown is allowed in: 00b Disabled (Link states shall be disabled for DMI PLL shutdown) 01b U0 or conditions for 10b setting. 10b U2 or conditions for 11b setting. 10b U3, Disconnected, Disabled or Powered-Off.
7:5	0h RW	**SS Backbone PXP PLL Shutdown Ux Enable (SSPLLSUE):** This register determines the Ux state(s) which will be exposed to DMI PLL Shutdown conditions. PLL Shutdown is allowed in: 000b Disabled (Link states shall be ignored for DMI PLL shutdown). 001b U0 or conditions for 010b setting 010b U1 or conditions for 011b setting 011b U2 or conditions for 100b setting 100b U3, Disconnected, Disabled or Powered-Off
4	0h RW	**XHC Backbone Local Clock Gating Enable (XHCBLCGE):** When set, this bit allows XHCI Controller IP backbone clock to be locally gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value.
3	0h RW	**HS Link Trunk Clock Gating Enable (HSLTCGE):** When set, this bit allows High Speed Link control's 480 MHz and its 48/60 MHz link clock trunk to be gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value. BIOS need to query fuses to see if HW is enabled.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**SS Link Trunk Clock Gating Enable (SSLTCGE):** When set, this bit allows the SuperSpeed Link control's 250 MHz and its divided 125 MHz link clock trunk to ge gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value. BIOS need to query fuses to see if HW is enabled.
1	0h RW	**IOSF Backbone Trunk Clock Gating Enable (IOSFBTCGE):** When set, this bit allows the IOSF backbone clock trunk to be gated when idle conditions are met.
0	0h RW	**IOSF Gasket Backbone Local Clock Gating Enable (IOSFBLCGE):** When set, this bit allows the IOSF Gasket backbone clock to be locally gated when idle conditions are met. Note: if XHC Dynamic Clock Gating Disable Fuse is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value. BIOS need to query fuses to see if HW is enabled.

16.967 Audio Time Synchronization (AUDSYNC_MMIO)—Offset 8658h

This 32 bit register is used for audio stream synchronization across different devices. Global signal sample_now captures a value in AUDSYNC register.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Rsvd2 (Rsvd2)**
29:16	0h RO/V	**Captured Frame List Current Index/Frame Number (CMFI):** The value in this register is updated in response to sample_now signal. Bits (29:16) reflect state of bits (13:0) of FRINDEX
15:13	0h RO	**Rsvd1 (Rsvd1)**
12:0	0h RO/V	**Captured Micro-frame BLIF (CMFB):** The value is updated in response to sample_now signal and provides information about offset within micro-frame. Captured value represents number of 8 high-speed bit time units from start of micro-frame. At the beginning of micro-frame captured value will be 0 and increase to maximum value at the end. Default maximum value is 7499 but it may be changed as result of adjustment done in FLA.

16.968 Serial Bus Release Number (SBRN_MMIO)—Offset 8660h

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 30h

Bit Range	Default & Access	Field Name (ID): Description
7:0	30h RO	**Serial Bus Release Number (SBRN):** A value of 30h indicates that this controller follows USB release 3.0.

16.969 Frame Length Adjustment (FLADJ_MMIO)—Offset 8661h

This feature is used to adjust any offset from the clock source that generates the clock that drives the SOF counter. When a new value is written into these six bits, the length of the frame is adjusted. Its initial programmed value is system dependent based on the accuracy of hardware USB clock and is initialized by system BIOS. This register should only be modified when the HChalted bit in the USBSTS register is a one. Changing value of this register while the host controller is operating yields undefined results.

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 60h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**Reserved (RSVD):** Read-Only. These bits are reserved for future use and should read as "00".
6	1h RO	**No Frame Length Timing Capability (NO_FRAME_LENGTH_TIMING_CAP):** This flag is set to 1 to indicate that the host controller does not support a programmable Frame Length Timing Value field.
5:0	20h RO	**Frame Length Timing Value (FLTV):** SOF micro-frame length) is equal to 59488 + value in this field. The default value is decimal 32 (20h), which gives a SOF cycle time of 60000. Frame Length (number of High Speed bit times) FLADJ Value (decimal) (decimal) 59488 0 (00h) 59504 1 (01h) 59520 2 (02h) ... 59984 31 (1Fh) 60000 32 (20h) ... 60480 62 (3Eh) 60496 63 (3Fh) Each decimal value change to this register corresponds to 16 high-speed bit times. The SOF cycle time (number of SOF counter clock periods to generate a SOF micro-frame length) is equal to 59488 + value in this field. The default value is decimal 32 (20h), which gives a SOF cycle time of 60000. Frame Length (# High Speed bit times) FLADJ Value

16.970 Best Effort Service Latency (BESL_MMIO)—Offset 8662h

Dummy register, mirror of physical register as BESL. Bset Effort Service Latency.

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:4	0h RW/L	**Default Best Effort Service Latency Deep (DBESLD):** Default Best Effort Service Latency (DBESLD) If the value of this field is non-zero, it defines the recommended value for programming the PORTPMSC register BESLD field. This is programmed by BIOS based on platform parameters.
3:0	0h RW/L	**Default Best Effort Service Latency (DBESL):** If the value of this field is non-zero, it defines the recommended value for programming the PORTPMSC register BESL field. This is programmed by BIOS based on platform parameters.

16.971 PCI Power Management Capability ID (PM_CID_MMIO)—Offset 8670h

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
7:0	1h RO	**PCI Power Management Capability ID (PM_CID):** A value of 01h indicates that this is a PCI Power Management capabilities field.

16.972 Next Item Pointer #1 (PM_NEXT_MMIO)—Offset 8671h

Dummy register, mirror of physical register as PM_NEXT. This register is modified and maintained by BIOS

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 80h

Bit Range	Default & Access	Field Name (ID): Description
7:0	80h RW/L	**Next Item Pointer #1 (PM_NEXT):** This register defaults to 80h, which indicates that the next capability registers begin at configuration offset 80h. This register is writable when the Access Control bit is set to '0'. This allows BIOS to effectively hide the next capability registers, if necessary. This register should only be written during system initialization before the plug-and-play software has enabled any master-initiated traffic. Values of: 80h implies next capability is MSI 00h implies that MSI capability is hidden. Note: This value is never expected to be programmed.

16.973 Power Management Capabilities (PM_CAP_MMIO)—Offset 8672h

Dummy register, mirror of physical register as PM_CAP. Normally, this register is read-only to report capabilities to the power management software. In order to report different power management capabilities depending on the system in which the Intel PCH is used, the write access to this register is controlled by the Access Control bit (ACCTRL). The value written to this register does not affect the hardware other than changing the value returned during a read. This register is modified and maintained by BIOS

Access Method

Type: MEM Register (Size: 16 bits)

Device:
Function:

Default: C1C2h

Bit Range	Default & Access	Field Name (ID): Description
15:11	18h RW/L	**PME_Support (PME_Support):** This 5-bit field indicates the power states in which the function may assert PME#. The Intel PCH XHC does not support the D1 or D2 states. For all other states, the Intel PCH XHC is capable of generating PME#. Software should never need to modify this field.
10	0h RW/L	**D2_Support (D2_Support):** The D2 state is not supported.
9	0h RW/L	**D1_Support (D1_Support):** The D1 state is not supported.
8:6	7h RW/L	**Aux_Current (Aux_Current):** The Intel PCH XHC reports 375mA maximum Suspend well current required when in the D3cold state. This value can be written by BIOS when a more accurate value is known.
5	0h RW/L	**DSI (DSI):** The Intel PCH reports 0, indicating that no device-specific initialization is required.
4	0h RO	**Reserved (RSVD)**

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW/L	**PME Clock (PMEClock):** The Intel PCH reports 0, indicating that no PCI clock is required to generate PME#.
2:0	2h RW/L	**Version (Version):** The Intel PCH reports 010, indicating that it complies with Revision 1.1 of the PCI Power Management Specification.

16.974 Power Management Control/Status (PM_CS_MMIO)—Offset 8674h

Dummy register, mirror of physical register as PM_CS

Access Method

Type: MEM Register (Size: 16 bits)

Device:

Function:

Default: 8h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**PME_Status (PME_Status):** This bit is set when the Intel PCH XHC would normally assert the PME# signal independent of the state of the PME_En bit. Writing a 1 to this bit will clear it and cause the internal PME to deassert (if enabled). Writing a 0 has no effect. This bit must be explicitly cleared by the operating system each time the operating system is loaded.
14:13	0h RO	**Data_Scale (Data_Scale):** The Intel PCH hardwires these bits to 00 because it does not support the associated Data register.
12:9	0h RO	**Data_Select (Data_Select):** The Intel PCH hardwires these bits to 0000 because it does not support the associated Data register.
8	0h RW	**PME_En (PME_En):** A 1 enables the Intel PCH XHC to generate an internal PME signal when PME_Status is 1. This bit must be explicitly cleared by the operating system each time it is initially loaded.
7:4	0h RO	**Reserved (RSVD)**
3	1h RO	**No Soft Reset (NSR):** No_Soft_Reset - When set ("1"), this bit indicates that devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset. Configuration Context is preserved. Upon transition from the D3hot to the D0 Initialized state, no additional operating system intervention is required to preserve Configuration Context beyond writing the PowerState bits. Transition from D3hot to D0 by a system or bus segment reset will return to the device state D0 Uninitialized with only PME context preserved if PME is supported and enabled.
2	0h RO	**Reserved (RSVD2)**

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**PowerState (PowerState):** This 2-bit field is used both to determine the current power state of XHC function and to set a new power state. The definition of the field values are: 00b - D0 state 11b - D3hot state If software attempts to write a value of 10b or 01b in to this field, the write operation must complete normally, however, the data is discarded and no state change occurs. When in the D3hot state, the Intel PCH must not accept accesses to the XHC memory range, but the configuration space must still be accessible.

16.975 Message Signaled Interrupt CID (MSI_CID_MMIO)—Offset 8680h

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 5h

Bit Range	Default & Access	Field Name (ID): Description
7:0	5h RO	**Capability ID (CID):** Indicates that this is an MSI capability

16.976 Next item pointer (MSI_NEXT_MMIO)—Offset 8681h

Dummy register, mirror of physical register as MSI_NEXT

Access Method

Type: MEM Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW/L	**Next Pointer (NEXT):** Indicates that this is the last item on the capability list

16.977 Message Signaled Interrupt Message Control (MSI_MCTL_MMIO)—Offset 8682h

Dummy register, mirror of physical register as MSI_MCTL

Access Method

Type: MEM Register (Size: 16 bits)
Device:
Function:

Default: 86h

Bit Range	Default & Access	Field Name (ID): Description
15:9	0h RO	**Reserved (RSVD)**
8	0h RO	**Per-Vector Masking Capable (PVM):** Specifies whether controller supports MSI per vector masking. Not supported
7	1h RO	**64 Bit Address Capable (C64):** Specifies whether capable of generating 64-bit messages. This device is 64-bit capable.
6:4	0h RW	**Multiple Message Enable (MME):** Indicates the number of messages the controller should assert. This device supports multiple message MSI.
3:1	3h RO	**Multiple Message Capable (MMC):** Indicates the number of messages the controller wishes to assert. This field must be set by HW to reflect the number of Interrupters supported. Encoding number of Vectors requested (number of Interrupters) 000 1 001 2 010 4 011 8 100 16 101 32 110-111 Reserved
0	0h RW	**MSI Enable (MSIE):** If set to 1, MSI is enabled and the traditional interrupt pins are not used to generate interrupts. If cleared to 0, MSI operation is disabled and the traditional interrupt pins are used.

16.978 Message Signaled Interrupt Message Address (MSI_MAD_MMIO)—Offset 8684h

Dummy register, mirror of physical register as MSI_MAD

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**Addr (Addr):** Lower DW of system specified message address, always DWORD aligned
1:0	0h RO	**Reserved (RSVD)**

16.979 Message Signaled Interrupt Upper Address (MSI_MUAD_MMIO)—Offset 8688h

Dummy register, mirror of physical register as MSI_MUAD

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Upper Addr (UpperAddr):** Upper DW of system specified message address.

16.980 Message Signaled Interrupt Message Data (MSI_MD_MMIO)—Offset 868Ch

Dummy register, mirror of physical register as MSI_MD

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW	**Data (Data):** This 16-bit field is programmed by system software if MSI is enabled. Its content is driven onto the lower word (PCI AD(15:0)) during the data phase of the MSI memory write transaction. The Multiple Message Enable field (bits 6-4 of the Message Control register) defines the number of low order message data bits the function is permitted to modify to generate its system software allocated vectors. For example, a Multiple Message Enable encoding of 010 indicates the function has been allocated four vectors and is permitted to modify message data bits 1 and 0 (a function modifies the lower message data bits to generate the allocated number of vectors). If the Multiple Message Enable field is 000, the function is not permitted to modify the message data.

16.981 Device Idle Capability (DEVIDLE_MMIO)—Offset 8690h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: F0140009h

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RO	**VID (VID)**

Bit Range	Default & Access	Field Name (ID): Description
27:24	0h RO	**REV (REV)**
23:16	14h RO	**Length (LENGTH):** Indicates that this capability is 16 bytes long.
15:8	0h RO	**Next Capability Pointer (NCP):** This field contains the offset to the next PCI Capability structure or 000h if no other items exist in the linked list of Capabilities.
7:0	9h RO	**Capability ID (CID)**

16.982 Vendor Specific Header (VSHDR_MMIO)—Offset 8694h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 1400010h

Bit Range	Default & Access	Field Name (ID): Description
31:20	14h RO	**VSEC Length (VSEC_LENGTH):** This field indicates the number of bytes in the entire VSEC structure, including the PCI Extended Capability header, the Vendor- Specific header, and the Vendor-Specific register
19:16	0h RO	**VSEC Rev (VSEC_REV):** This field is a vendor-defined version number that indicates the version of the VSEC structure. Software must qualify the Vendor ID and VSEC ID before interpreting this field.
15:0	10h RO	**VSEC ID (VSEC_ID):** This field is a vendor-defined ID number that indicates the nature and format of the VSEC structure. Software must qualify the Vendor ID before interpreting this field.

16.983 SW LTR POINTER (SWLTRPTR_MMIO)—Offset 8698h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**SW LTR Update MMIO Offset Location (SW_LTR_UPDT_MMIO_OFFSET):** This register contains the location pointer to the SW LTR register in MMIO space, as an offset from the specified BAR. The value of this register is a don't care, if the Valid bit is not set.
3:1	0h RO	**Base Address Register Number (BARNUM):** Contains the 0's based AR Number of the BAR which contains the location of the SW LTR MMIO register. When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number. Bar 0 is the 1st BAR. The value of this register is a don't care, if the Valid bit is not set.
0	0h RO	**Valid (VALID):** Set to '1' to indicate that the function has implemented a SW LTR register as specified in the DevIdle that can be located using the SWLTRLOC register and BARNUM. Set to '0' to indicate that the function has not implemented a SW LTR register that is compliant to the DevIdle definition. This could be because the function has not implemented SW LTR at all, or has a device specific version of SW LTR.

16.984 Device Idle Pointer Register (DEVIDLEPTR_MMIO)—Offset 869Ch

Dummy register, mirror of physical register as DEVIDLEPTR

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 80AC1h

Bit Range	Default & Access	Field Name (ID): Description
31:4	80ACh RW/L	**DevIdle MMIO Offset Location (DEVIDLELOC):** This register contains the location pointer to the DevIdle register in MMIO space, as an offset from the specified BAR. The value of this register is a don't care, if the Valid bit is not set.
3:1	0h RO	**Base Address Register Number (BARNUM):** Contains the 0's based BAR Number of the BAR which contains the location of the DevIdle MMIO register. When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number. Bar 0 is the 1st BAR. The value of this register is a don't care, if the Valid bit is not set.

Bit Range	Default & Access	Field Name (ID): Description
0	1h RW/L	**Valid (VALID):** Set to '1' to indicate that the function has implemented a DevIdle register as specified in the DevIdle that can be located using the DEVIDLELOC register and BARNUM. Set to '0' to indicate that the function has not implemented a DevIdle register that is compliant to the DevIdle definition. This could be because the function has not implemented DevIdle at all, or has a device specific version of DevIdle.

16.985 Device Idle Power ON Latency (DEVIDLEPOL_MMIO)—Offset 86A0h

Dummy register, mirror of physical register as DEVIDLEPOL

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 800h

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	**Rsvd1 (Rsvd1)**
12:10	2h RW/L	**Power On Latency Scale (POLS):** Latency Scale multiplier: 010: 1us 011: 32us All other settings are reserved. The value of this register is multiplied with the Power On Latency Value (POLV) to provide the DevIdle power on exit latency multiplier scale from 1us to 32ms. This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value. This register is undefined if the DevIdle.Valid bit is '0'.
9:0	0h RW/L	**Power On Latency Value (POLV):** 10-bit value that is multiplied by the Power On Latency Scale. A value of 0 indicates a power on latency of less than 1us. This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value. This register is undefined if the DevIdle.Valid bit is '0'.

16.986 High Speed Configuration 2 (HSCFG2_MMIO)—Offset 86A4h

Dummy register, mirror of physical register as HSCFG2

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2000h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Rsvd1 (Rsvd1)**
18	0h RW	**PORT1_HOST_MODE_OVERRIDE (PORT1_HOST_MODE_OVERRIDE):** When set, this bit causes the Host_Device mux on port 1 to be forced into the Host mode.
17:16	0h RW	**eUSB2SEL (eUSB2SEL):** The two bits are associate with USB2 ports 1 - bit 16 and 2 - bit 2 0: Port is mapped to USB2 1: Port is mapped to eUSB2
15	0h RW	**HS ASYNC Active IN Mask (HSAAIM):** Determines if the Async Active will mask/ignore IN EP s. 0 HS ASYNC Active will include IN EP s. 1 HS ASYNC Active will mask/ignore IN EP s.
14	0h RW	**HS OUT ASYNC Active Polling EP Mask (HSOAAPEPM):** Determines if the Async Active for OUT HS/FS/LS masks/ignores EP s that are polling/PINGing (HS) due to NAK. 0 HS OUT ASYNC Active will include EP s that are polling. 1 HS OUT ASYNC Active will mask/ignore EP s that are polling.
13	1h RW	**HS IN ASYNC Active Polling EP Mask (HSIAAPEPM):** Determines if the Async Active for IN HS/FS/LS masks/ignores EP s that are polling due to NAK. 0 HS IN ASYNC Active will include EP s that are polling. 1 HS IN ASYNC Active will mask/ignore EP s that are polling.
12:11	0h RW	**HS INTR IN Periodic Active Policy Control (HSIIPAPC):** Controls how the HS INTR IN periodic active is used to generate the global periodic active. This will determine how the smallest service interval among active EP s and number of active EP s are used. 0 HS INTR IN periodic active will be used to generate periodic active if Service Interval Threshold OR Numb of EP Threshold values meet the requirement. 1 HS INTR IN periodic active will be used to generate periodic active if Service Interval Threshold AND Numb of EP Threshold values meet the requirement. 2 Always allow HS INTR EP s to be used in the generation of the global Periodic Active indication. 3 Never allow HS INTR EP s to be used in the generation of the global Periodic Active indicaiton
10:4	0h RW	**HS INTR IN Periodic Active Num of EP Threshold (HSIIPANEPT):** Defines the threshold used to determine if Periodic Acive may include HS/FS/LS INTR IN EP active indication. If there are more than NumEPThreshold active HS/FS/LS INTR EP s then they may be included as part of the periodic active generation.
3:0	0h RW	**HS INTR IN Periodic Active Service Interval Threshold (HSIIPASIT):** Defines the Service Interval threshold used to determine if Periodic Acive will include HS/FS/LS INTR IN EP active indication. If there are any active HS/FS/LS INTR EP s with a service interval less than or equal to this threshold then they may be included as part of the periodic active generation.

16.987 XHCI USB2 Overcurrent Pin Mapping 1 (U2OCM1_MMIO)—Offset 86B0h

The RW/L property of this register is controlled by OCCFDONE bit.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Rsvd (Rsvd):** Reserved
8:0	0h RW/L	**OC Mapping (OCM)**

16.988 XHCI USB2 Overcurrent Pin Mapping 2 (U2OCM2_MMIO)—Offset 86B4h

Reserved

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Rsvd (Rsvd):** Reserved
8:0	0h RW/L	**OC Mapping (OCM)**

16.989 XHCI USB3 Overcurrent Pin Mapping 1 (U3OCM1_MMIO)—Offset 86D0h

The RW/L property of this register is controlled by OCCFDONE bit.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Rsvd (Rsvd)**
6:0	0h RW/L	**OC Mapping (OCM)**

16.990 XHCI USB3 Overcurrent Pin Mapping 2 (U3OCM2_MMIO)—Offset 86D4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Rsvd (Rsvd)**
6:0	0h RW/L	**OC Mapping (OCM)**

16.991 XHCC3 (XHCC3_MMIO)—Offset 86FCh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RO	**Reserved (Rsvd1):** Reserved
4	0h RW	**Error Handing: Disable masking of extra read to completion collect buffer (SUPP_XTRA_RD_CPL_COLL_BUF):** When set to 1, the fix to supress extra read to completion collect buffer is disabled. The fix is relevent only when ECC checking is enabled
3	0h RW	**Error Handling : Disable Command Parity Check (DISABLE_COMMAND_PARITY_CHECK):** When set to 1, XHCI Host Controller disables checking of Command Parity on command received as a target on its IOSF Primary interface
2	0h RW	**Error Handling : Disable Data Parity Check (DISABLE_DATA_PARITY_CHECK):** When set to 1, XHCI Host Controller disables checking of Data Parity on data received as a target on its IOSF Primary interface

Bit Range	Default & Access	Field Name (ID): Description
1	1h RW	**Error Handling : Enable ECC Error Response (ENABLE_ECC_ERROR_RESPONSE):** When set to 1, XHCI Host Controller will check for ECC on RFs (that support ECC) and halt operation when uncorrectable ECC is detected
0	0h RW/L	**Function Disable (FXN_DISABLE):** When set will disable the xHC from being operational.

16.992 Debug Capability ID Register (DCID)—Offset 8700h

This register is modified and maintained by BIOS

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 5100Ah

Bit Range	Default & Access	Field Name (ID): Description
31:21	0h RO	**Reserved (RSVD)**
20:16	5h RW	**Debug Capability Event Ring Segment Table Max (DCERSTM):** Note: This register is sticky.
15:8	10h RW	**Next Capability Pointer (NCP):** Note: This register is sticky.
7:0	Ah RW	**Capability ID (CID):** Note: This register is sticky.

16.993 Debug Capability Doorbell Register (DCDB)—Offset 8704h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Reserved (RSVD)**
15:8	0h RW	**Doorbell Target (DBTGT):** This field defines the target of the doorbell reference. The table below defines the Debug Capability notification that is generated by ringing the doorbell. Value Definition 0 Data EP 1 OUT Enqueue Pointer Update 1 Data EP 1 IN Enqueue Pointer Update 2:255 Reserved This field returns '0' when read and the value should be treated as undefined by software.

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Reserved (RSVD_1)**

16.994 Debug Capability Event Ring Segment Table Size Register (DCERSTSZ)—Offset 8708h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Reserved (RSVD)**
15:0	0h RW	**Event Ring Segment Table Size (ERSTS):** This field identifies the number of valid Event Ring Segment Table entries in the Event Ring Segment Table pointed to by the Debug Capability Event Ring Segment Table Base Address register. The maximum value supported by an xHC implementation for this register is defined by the DCERST Max field in the DCID register Software shall initialize this register before setting the Debug Capability Enable field in the DCCTRL register to '1'.

16.995 Debug Capability Event Ring Segment Table Base Address Register (DCERSTBA)—Offset 8710h

Access Method

Type: MEM Register
(Size: 64 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:4	0h RW	**Event Ring Segment Table Base Address Register (ERSTBAR):** This field defines the high order bits of the start address of the Debug Capability Event Ring Segment Table. Software shall initialize this register before setting the Debug Capability Enable field in the DCCTRL register to '1'.
3:0	0h RW	**Reserved (RSVD)**

16.996 Debug Capability Event Ring Dequeue Pointer Register (DCERDP)—Offset 8718h

Access Method

Type: MEM Register (Size: 64 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:4	0h RW	**Dequeue Pointer (DQP):** This field defines the high order bits of the 64-bit address of the current Debug Capability Event Ring Dequeue Pointer. Software shall initialize this register before setting the Debug Capability Enable field in the DCCTRL register to '1'.
3	0h RW	**Reserved (RSVD)**
2:0	0h RW	**Dequeue ERST Segment Index (DESI):** This field may be used by the xHC to accelerate checking the Event Ring full condition. This field is written with the low order 3 bits of the offset of the ERST entry which defines the Event Ring segment that the Event Ring Dequeue Pointer resides in.

16.997 Debug Capability Control Register (DCCTRL)—Offset 8720h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Debug Capability Enable (DCE)**
30:24	0h RO	**Device Address (DADDR)**
23:16	0h RO	**Debug Max Burst Size (DMBS):** LPT-LP USB Debug Device does not support bursting.
15:5	0h RO	**Reserved (RSVD)**
4	0h RW/C	**DbC Run Change (DRC)**
3	0h RW/1S	**Halt IN TR (HIT)**
2	0h RW/1S	**Halt OUT TR (HOT)**

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Link Status Event Enable (LSE)**
0	0h RO	**DbC Run (DCR)**

16.998 Debug Capability Status Register (DCST)—Offset 8724h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Debug Port Number (DPNUM):** This field provides the ID of the Root Hub port that the Debug Capability has been automatically attached to. The value is '0' when the Debug Capability is not attached to a Root Hub port.
23:1	0h RO	**Reserved (RSVD)**
0	0h RO	**Event Ring Not Empty (ERNE):** When '1', this field indicates that the Debug Capability Event Ring has a Transfer Event on it. It is automatically cleared to '0' by the xHC when the Debug Capability Event Ring is empty, i.e. the Debug Capability Enqueue Pointer is equal to the Debug Capability Event Ring Dequeue Pointer register.

16.999 Debug Capability Port Status and Control Register (DCPORTSC)—Offset 8728h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 80h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD)**
23	0h RW/C	**Port Config Error Change (CEC):** This flag indicates that the port failed to configure its link partner. 0 = No change. 1 = Port Config Error detected. Software shall clear this bit by writing a '1' to it.

Bit Range	Default & Access	Field Name (ID): Description
22	0h RW/C	**Port Link Status Change (PLC):** This flag is set to '1' due to the following PLS transitions: U0 -) U3 Suspend signaling detected from Debug Host U3 -) U0 Resume complete Polling -) Disabled Training Error Ux or Recovery -) Inactive Error Software shall clear this bit by writing a '1' to it. This field is '0' if DCE is '0'
21	0h RW/C	**Port Reset Change (PRC):** This bit is set when reset processing on this port is complete (i.e. a '1' to '0' transition of PR). '0' = No change. '1' = Reset complete.Software shall clear this bit by writing a '1' to it. This field is '0' if DCE is '0'.
20:18	0h RO	**Reserved (RSVD_1)**
17	0h RW/C	**Connect Status Change (CSC):** '1' = Change in Current Connect Status. '0' = No change. Indicates a change has occurred in the port's Current Connect Status. The xHC sets this bit to '1' for all changes to the Debug Device connect status, even if system software has not cleared an existing DbC Connect Status Change. For example, the insertion status changes twice before system software has cleared the changed condition, hardware will be "setting" an already-set bit (i.e., the bit will remain '1'). Software shall clear this bit by writing a '1' to it. This field is '0' if DCE is '0'.
16:14	0h RO	**Reserved (RSVD_2)**
13:10	0h RO	**Port Speed (PSPD):** This field identifies the speed of the port. This field is only relevant when a Debug Host is attached (CCS = '1') in all other cases this field shall indicate Undefined Speed. 0 Undefined Speed 1-15 Protocol Speed ID (PSI) Note: The Debug Capability onlydoes not supports LS, FS, or HS operation.
9	0h RO	**Reserved (RSVD_3)**
8:5	4h RO	**Port Link State (PLS):** This field reflects its current link state. This field is only relevant when a Debug Host is attached (Debug Port Number) '0'). Value Meaning 0 Link is in the U0 State 1 Link is in the U1 State 2 Link is in the U2 State 3 Link is in the U3 State (Device Suspended) 4 Link is in the Disabled State 5 Link is in the RxDetect State 6 Link is in the Inactive State 7 Link is in the Polling State 8 Link is in the Recovery State 9 Link is in the Hot Reset State 15:10 Reserved Note: Transitions between different states are not reflected until the transition is complete.
4	0h RO	**Port Reset (PR):** '1' = Port is in Reset. '0' = Port is not in Reset. This bit is set to '1' when the bus reset sequence as defined in the USB Specification is detected on the Root Hub port assigned to the Debug capability. It is cleared when the bus reset sequence is completed by the Debug Host, and the DbC shall transition to the USB Default state. A '0' to '1' transition of this bit shall clear DCPORTSC PED ('0'). This field is '0' if DCE or CCS are '0'.
3:2	0h RO	**Reserved (RSVD_4)**

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Port Enabled/Disabled (PED):** Default = '0'. '1' = Enabled. '0' = Disabled. This flag shall be set to '1' by a '0' to '1' transition of CCS or a '1' to '0' transition of the PR. When PED transitions from '1' to '0' due to the assertion of PR, the port's link shall transition to the Rx.Detect state. This flag may be used by software to enable or disable the operation of the Root Hub port assigned to the Debug Capability. The Debug Capability Root Hub port operation may be disabled by a fault condition (disconnect event or other fault condition, e.g. a LTSSM Polling substate timeout, tPortConfiguration timeout error, etc.), the assertion of DCPORTSC PR, or by software. 0 = Debug Capability Root Hub port is disabled. 1 = Debug Capability Root Hub port is enabled. When the port is disabled (PED = '0') the port's link shall enter the SS.Disabled state and remain there until PED is reasserted ('1') or DCE is negated ('0'). Note that the Root Hub port is remains mapped to Debug Capability while PED = '0'. While PED = '0' the Debug Capability will appear to be disconnected to the Debug Host. Note, this bit is not affected by PORTSC PR bit transitions. This field is '0' if DCE or CCS are '0'.
0	0h RO	**Current Connect Status (CCS):** '1' = A Root Hub port is connected to a Debug Host and assigned to the Debug Capability. '0' = No Debug Host is present. This value reflects the current state of the port, and may not correspond to the value reported by the Connect Status Change (CSC) field in the Port Status Change Event that was generated by a '0' to '1' transition of this bit. This flag is '0' if Debug Capability Enable (DCE) is '0'.

16.1000 Debug Capability Context Pointer Register (DCCP)—Offset 8730h

Access Method

Type: MEM Register (Size: 64 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:4	0h RW	**Debug Capability Context Pointer Register (DCCPR):** This field defines the high order bits of the start address of the Debug Capability Context data structure associated with the Debug Capability. Software shall initialize this register before setting the Debug Capability Enable bit in the Debug Capability Control Register to '1'.
3:0	0h RO	**Reserved (RSVD)**

16.1001 Debug Capability Device Descriptor Info Register 1 (DCDDI1)—Offset 8738h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 80870000h

Bit Range	Default & Access	Field Name (ID): Description
31:16	8087h RW	**Vendor ID (VID):** This field is presented by the Debug Device in the USB Device Descriptor idVendor field.
15:8	0h RO	**Reserved (RSVD)**
7:0	0h RW	**DbC Protocol (DBCPR):** This field is presented by the Debug Device in the USB Interface Descriptor bInterfaceProtocol field. Value Function 0 Debug Target vendor defined. 1 GNU Remote Debug Command Set supported. 2-255 Reserved.

16.1002 Debug Capability Device Descriptor Info Register 2 (DCDDI2)—Offset 873Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Device Revision (DREV):** This field is presented by the Debug Device in the USB Device Descriptor bcdDevice field.
15:0	0h RW	**Product ID (PID):** This field is presented by the Debug Device in the USB Device Descriptor idProduct field.

16.1003 Debug Capability Descriptor Parameters (DCDP)—Offset 8740h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 30C3h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD)**
23:16	0h RW	**Max Power Field (MPF):** This field will be used by USB Debug Device to report maximum power consumption when the device is fully operational. This value is returned by bMaxPower field in response to Configuration Descriptor read from the debug device. Note: bU1DevExitLat and bU2DevExitLat fields returned in BOS Descriptor read will be taken from the corresponding fields from the Host Controller space.
15:8	30h RW/S	**NEXT CAPABILITY POINTER (NCP)**
7:0	C3h RW/S	**Capability ID (CID)**

16.1004 Debug Device Control ODMA (DBGDEV_CTRL_ODMA_REG)—Offset 8748h

This register contains a number of fields that provide a specific level of configurability for the OUT DMA that is part of Debug Device logic. This configurability is above and beyond that defined in the xHCI specification.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RW	**Reserved (RSVD)**
18	0h RW	**Enable ACK FIFO credit accounting (EN_ACK_FCA):** Setting this field will enable ACK FIFO credit accounting. ODMA will ensure that ample room exists in the ACK FIFO for expected device responses prior to initiating a given DP
17:14	0h RW	**Reserved (RSVD_1)**
13	0h RW	**Enable ACK FIFO ICA mechanisms (EN_ACK_FIFO_ICA):** Setting this field will enable ACK FIFO individual credit accounting mechanisms for Async vs. Periodic Endpoints. ODMA will ensure that ample room exists in the ACK FIFO for expected device responses prior to initiating a given DP
12:9	0h RW	**Reserved (RSVD_2)**

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW	**Clear ownership of context semaphore (CL_OWN_CS):** Setting this field generates a pulse that clears the ownership of the context semaphore that is shared between the Out DMA Response and Completion Finite State Machines
7	0h RW	**Return OD ACK credits (RET_OD_ACK_CR):** Setting this field generates a pulse that implicitly returns all of the Out DMA ACK credits on all ports
6	0h RW	**Reserved (RSVD_3)**
5	0h RW	**Return ODCF SM to idle state (RET_ODCF_SM_IS):** Setting this field generates a pulse that returns the Out DMA Completion Finite State Machine into the IDLE state
4	0h RW	**Return ODRF SM to idle state (RET_ODRF_SM_IS):** Setting this field generates a pulse that returns the Out DMA Response Finite State Machine into the IDLE state
3	0h RW	**Return ODRDF SM to idle state (RET_ODRDF_SM_IS):** Setting this field generates a pulse that returns the Out DMA Read Finite State Machine into the IDLE state
2:0	0h RW	**Reserved (RSVD_4)**

16.1005 DBC Control Register 1 (DBCCTL_REG)—Offset 8760h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD):** RSVD
7	0h RW	**SW_DCE_SEL (SW_DCE_SEL)**
6:3	0h RW	**DISC_RXD_CNT (DISC_RXD_CNT)**
2	0h RO	Reserved.
1	0h RW	**Force DCE Mode (FORCE_DCE_MODE):** 0: When DCE is set, the DbC switches to Mode 2 1: When DCE is set, the DbC switches to Mode 3

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**Force Disconnect upon DCE (FORCE_DISCONNECT_ON_DCE):** If this bit is set by BIOS, the DbC will temporarily disconnect from the remote host if the DCE is set, and shortly thereafter re-connect. This allows the DbC to switch from Mode1 to Mode2 or Mode 3 operation upon DCE being set.

16.1006 SSIC Policy and Implementation Specific Registers Capability ID register (SSIC_IMPLEMENTATION_CAPABILITY_ID_REG)—Offset 8800h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 40C4h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD)**
15:8	40h RO	**Next Capability Pointer (NCP)**
7:0	C4h RO	**Supported Protocol ID (PID)**

16.1007 SSIC global configuration control register (SSIC_GLOBAL_CONFIG_CONTROL_REG)—Offset 8804h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Reserved (RSVD)**
17:16	0h RW	**T_ACT_H8_MARGIN (T_ACT_H8_MARGIN):** Specifies the margin time added to the tActivate spec time of 100us in T_ACT_H8_TIME. 11: Reserved 10: 50us 01: 20us (default for ModPHY) 00: 10us (default for MIPI MPHY)
15:4	0h RO	**Reserved1 (RSVD1)**

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**BB_PLL_OVRD_DURING_PWM (BB_PLL_OVRD_DURING_PWM)**
2	0h RW	**CMN_LANE_PWRGATE_DIS (CMN_LANE_PWRGATE_DIS):** This will disable the MPHY common lane power gate.
1	0h RW	**HS_CLK_GATE_DIS (HS_CLK_GATE_DIS):** This will disable the HS Clock gate request from XHCI to MIPI PLL.
0	0h RW	**PWM_CLK_GATE_DIS (PWM_CLK_GATE_DIS):** This will disable the PWM Clock gate request from XHCI to MIPI PLL.

16.1008 (PORT1_SSIC_CONFIG_REG1)—Offset 8808h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 82286010h

Bit Range	Default & Access	Field Name (ID): Description
31:30	2h RW	**NUM_RRAP_ATTEMPT (NUM_RRAP_ATTEMPT):** Specify the number of RRAP access attempt if there was tRRAPInitiatorResponse timeout.
29:22	8h RW	**TX_MIN_STALL (TX_MIN_STALL):** (TX_Min_STALL_NoConfig_Time_Capability in MIPI PHY Spec). PA will ensure TX_MIN_STALL time in STALL before entering HS-BURST or HIBERN8. Specifies minimum time in SI in STALL state.
21:17	14h RW	**PWM_EXIT_TIME (PWM_EXIT_TIME):** This will be max of RX_Min_ActivateTime_Capability and TX_Min_SAVE_Config_Time_Capability in MIPI PHY Specs) Specifies minimum activate time needed in 5us steps. PA will wait PWM_EXIT_TIME time after exiting PWM to start HS-BURST. SSIC spec requires it to be at 100us. For speedup mode this time would be in 1us steps.
16	0h RW	**DSP_DISC_BURST_CLOSE_RRAP (DSP_DISC_BURST_CLOSE_RRAP):** 0 - Send BURST Closure RRAP for DSP Disconnect 1 - Dont send Burst Closure RRAP for DSP Disconnect. For device PWM exit would be indication for DSP Disconnect.
15:13	3h RW	**TX_MIN_ACTIVATE (TX_MIN_ACTIVATE):** (TX_Min_ActivateTime in MIPI PHY Spec) Specifies minimum activate time needed in 100us steps. PA will wait TX_MIN_ACTIVATE time after H8 exit to start of PWM Burst. SSIC spec requires it to be at 1.5ms. For speedup mode this time would be 10us to 150us in steps of 10us.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**RRAP_BYPASS (RRAP_BYPASS):** Enables DFx Loopback test mode where RRAP command /response is bypassed for HS_CONFIG and PWM_BURST_CLOSURE. Host will still perform initial enter and exit PWM mode with no RRAP transfers.
11	0h RW	**Reserved (RSVD)**
10:8	0h RW	**T_ACT_H8_EXIT (T_ACT_H8_EXIT):** Specifies minimum time in 100us steps for T_ACTIVATE_TIME to exit H8. Val: Si Time / Fast Sim Time 000: 110us / 10us 001: 210us / 20us 110: 710us / 70us 111: 5ms / 500us Spec required timing is 100us.
7:5	0h RW	**MIN_HIBERN8_TIME (MIN_HIBERN8_TIME):** Minimum time in HIBERN8 state 0 110us 1 210us .. 7 810us PA will ensure MIN_HIBERN8_TIME in its TX HIBERN8. For speedup mode this time will be from 10us to 80us in steps of 10us.
4	1h RW	**SSICRATE (SSICRATE):** 0 B Series 1 A Series
3:2	0h RW	**HSGEAR (HSGEAR):** 0=HS-G1 1=HS-G2 2=HS-G3
1:0	0h RW	**SSICLANE (SSICLANE):** 0=Single lane 1=Bi-lane 2=Quad-lane

16.1009 (PORT1_SSIC_CONFIG_REG2)—Offset 880Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 4881F46h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**SSIC_PORT_UNUSED (SSIC_PORT_UNUSED):** This indicates that no SSIC device is connected to the port and PORTSC register would always reflect DISCONNECTED.
30	0h RW	**PROG_DONE (PROG_DONE):** BIOS will program this bit once SSIC profile programming is done. SSIC will enable the MPHY and local PLL once this bit is set.
29:26	1h RW	**NUM_OF_MK0 (NUM_OF_MK0):** Number of MK0 which transmitter will send before sending any data.
25	0h RW	**DISABLE_SCRAMBLING (DISABLE_SCRAMBLING):** This bit will disable scrambling in HS-BURST
24:21	4h RW	**RETRAIN_TIME (RETRAIN_TIME):** Corresponds to time in 10us to detect improper training of the local and remote M-RX as part of HS-BURST entry. SSIC Spec specific 40 to 50us
20:16	8h RW	**PHY_RESET_TIME (PHY_RESET_TIME):** Corresponds to time in 100ns, PA will drive PHY RESET for MIPI PHY.

Bit Range	Default & Access	Field Name (ID): Description
15:8	1Fh RW	**LRST_TIME (LRST_TIME):** Corresponds to time in 100us PA will drive DIF-P for line reset. MIPI PHY specifies 3.1 ms Minimum. For speedup mode this time will be in steps of 5us.
7:0	46h RW	**ACTIVATE_LRST_TIME (ACTIVATE_LRST_TIME):** (Corresponds to tResetDIFN) Specifies in step of 1ms period of time a DSP is required to drive a DIF-N prior to a LINE-RESET. SSIC spec defined range is 60-80ms. For speedup mode this time will be from 5us to 400us in steps of 5us.

16.1010 (PORT1_SSIC_CONFIG_REG3)—Offset 8810h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 40400000h

Bit Range	Default & Access	Field Name (ID): Description
31:28	4h RW	**U0_STALL_TO (U0_STALL_TO):** Time in 2^[U-STALL_TO] clock for U0 STALL entry. If there is no packet in arbiter for U0_STALL_TO time, PA will enable U0 STALL if DISABLE_U0_STALL is not set. 0000 0011 (Not recommended) 0100 16 SSIC HS CLK . 1111 32768 SSIC HS CLK
27	0h RW	**MPHY_TEST_MODE_EN (MPHY_TEST_MODE_EN):** When this bit is set, controller would not initiate any PWM once it enters PWM Mode. It will wait for tester to send Loopback RRAP Command.
26	0h RW	**DL_PWR_GATE_DIS (DL_PWR_GATE_DIS)**
25:21	2h RW	**HIBERN8_ENTER_TX (HIBERN8_ENTER_TX):** The time PA will drive DIF-N after last bit of Line-CFG before entering H8. In steps of 50ns. Legal range is 50 to 1000ns.
20:19	0h RW	**LUP_LDN_TIMER_MAX (LUP_LDN_TIMER_MAX):** 00: 10 us 01: 100 us 10: 1000us /1ms 11: (Timer is disable and dont transmit LDN)
18:3	0h RW	**RESERVED (RSVD):** rsvd
2	0h RW	**DISABLE_U0_STALL (DISABLE_U0_STALL):** This bit will disable the STALL entry in U0
1	0h RW	**SSIC_PG_U3_DIS (SSIC_PG_U3_DIS):** This disables MPHY DL PG during U3.
0	0h RW	**SSIC_PG_U2_DIS (SSIC_PG_U2_DIS):** This disables MPHY DL PG during U2.

16.1011 (PORT1_SSIC_CONFIG_REG4)—Offset 8814h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 64796464h

Bit Range	Default & Access	Field Name (ID): Description
31:24	64h RW	**PM_LC_MAX_TIMER (PM_LC_MAX_TIMER):** In steps of 1us. Default 100us
23:16	79h RW	**PM_ENTRY_TIMER_MAX (PM_ENTRY_TIMER_MAX):** In steps of 1us. Default 100us
15:8	64h RW	**PEND_HP_TIMER_MAX (PEND_HP_TIMER_MAX):** In steps of 1us. Default 100us
7:0	64h RW	**CRD_PEND_TIMER_MAX (CRD_PEND_TIMER_MAX):** In steps of 1us. Default 100us

16.1012 (PORT1_SSIC_LOOPBACK_CONFIG_REG)—Offset 8818h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**RESERVED (RSVD)**
7:4	0h RW	**LOOPBACK_EN (LOOPBACK_EN):** This Register is used to enable Conformance Loopback in the Target in the specified PAIR.. Bit [0]: Writing 1b1 enables the loopback mode in PAIR0. Writing 1b0 shall have no effect. Bit [1]: Writing 1b1 enables the loopback mode in PAIR1. Writing 1b0 shall have no effect. Bit [2]: Writing 1b1 enables the loopback mode in PAIR2. Writing 1b0 shall have no effect. Bit [3]: Writing 1b1 enables the loopback mode in PAIR3. Writing 1b0 shall have no effect.
3:0	0h RW	**RX LOOPBACK CNTR RESET (RX_LOOPBACK_CNTR_RESET):** Reset RX_BURST_COUNT and RX_ERR_COUNT, write only, self-clearing. Bit [0] for PAIR0 Bit [1] for PAIR1 Bit [2] for PAIR2 Bit [3] for PAIR3

16.1013 (PORT1_SSIC_LOOPBACK_BURST_COUNT_REG)—Offset 881Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**RX_BURST_COUNT_LANE3 (RX_BURST_COUNT_LANE3):** RX_BURST_COUNT shall increment each time the M-RX of the PAIR under test detects a prepare time DIF-P. This shall be reset when the DUT receives a Write Command with rx_count_reset set to one. These counters shall not be reset when the DUT receives a LINE-RESET.
23:16	0h RO	**RX_BURST_COUNT_LANE2 (RX_BURST_COUNT_LANE2):** RX_BURST_COUNT shall increment each time the M-RX of the PAIR under test detects a prepare time DIF-P. This shall be reset when the DUT receives a Write Command with rx_count_reset set to one. These counters shall not be reset when the DUT receives a LINE-RESET.
15:8	0h RO	**RX_BURST_COUNT_LANE1 (RX_BURST_COUNT_LANE1):** RX_BURST_COUNT shall increment each time the M-RX of the PAIR under test detects a prepare time DIF-P. This shall be reset when the DUT receives a Write Command with rx_count_reset set to one. These counters shall not be reset when the DUT receives a LINE-RESET.
7:0	0h RO	**RX_BURST_COUNT_LANE0 (RX_BURST_COUNT_LANE0):** RX_BURST_COUNT shall increment each time the M-RX of the PAIR under test detects a prepare time DIF-P. This shall be reset when the DUT receives a Write Command with rx_count_reset set to one. These counters shall not be reset when the DUT receives a LINE-RESET.

16.1014 (PORT1_SSIC_LOOPBACK_ERROR_COUNT_REG)—Offset 8820h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**RX_ERR_COUNT_LANE3 (RX_ERR_COUNT_LANE3):** RX_BURST_COUNT shall increment each time the M-RX of the PAIR under test detects a prepare time DIF-P. This shall be reset when the DUT receives a Write Command with rx_count_reset set to one. These counters shall not be reset when the DUT receives a LINE-RESET.

Bit Range	Default & Access	Field Name (ID): Description
23:16	0h RO	**RX_ERR_COUNT_LANE2 (RX_ERR_COUNT_LANE2):** RX_BURST_COUNT shall increment each time the M-RX of the PAIR under test detects a prepare time DIF-P. This shall be reset when the DUT receives a Write Command with rx_count_reset set to one. These counters shall not be reset when the DUT receives a LINE-RESET.
15:8	0h RO	**RX_ERR_COUNT_LANE1 (RX_ERR_COUNT_LANE1):** RX_BURST_COUNT shall increment each time the M-RX of the PAIR under test detects a prepare time DIF-P. This shall be reset when the DUT receives a Write Command with rx_count_reset set to one. These counters shall not be reset when the DUT receives a LINE-RESET.
7:0	0h RO	**RX_ERR_COUNT_LANE0 (RX_ERR_COUNT_LANE0):** RX_BURST_COUNT shall increment each time the M-RX of the PAIR under test detects a prepare time DIF-P. This shall be reset when the DUT receives a Write Command with rx_count_reset set to one. These counters shall not be reset when the DUT receives a LINE-RESET.

16.1015 SSIC Local and Remote Profile Registers Capability ID register (SSIC_PROFILE_CAPABILITY_ID_REG)—Offset 8900h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: FFC5h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD)**
15:8	FFh RO	**Next Capability Pointer (NCP)**
7:0	C5h RO	**Supported Protocol ID (PID)**

16.1016 SSIC Port N Register Access Control (PORT1_REGISTER_ACCESS_CONTROL)—Offset 8904h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: C00000h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (RSVD)**
25	0h RW	**RRAP Register Bank Valid (RRAP_REG_BANK_VALID):** 0 = No valid Commands in the Lane N register bank. Host Controller can close the PWM burst once commands from this register are completed. 1 = Valid commands present in the Lane N register bank. Host controller must complete commands from the register bank before closing the burst.
24	0h RW	**Target Phy (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy Setting this bit to 1 allows the use of this command mechanism to write to local Phy profile and AFE tuning registers Primarily as a back up option.
23	1h RW	**HS_Config (HS_CONFIG):** When this bit is set to 1 the host controller will issue an RRAP write with HS_Config=1 once it sees Command Phase Done = 1
22	1h RW	**Command Phase Done (CPD):** When set to 1, this indicates that SW has completed performing RRAP cycles through the command register.
21	0h RW	**Command Valid (CMD_VALID):** When written to 1 indicates that the Attribute ID and Attribute Data for writes fields are valid.
20	0h RW	**Read_Write (READ_WRITE):** 0 = Write 1= Read
19:8	0h RW	**Attribute ID (ATT_ID):** Attribute ID that is being written or read
7:0	0h RW	**Attribute Write Data (ATT_WRITE_DATA):** Data byte that is required to be written to either the local phy or the remote phy

16.1017 SSIC Port N Register Access Status (PORT1_REGISTER_ACCESS_STATUS)—Offset 8908h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RO	**Reserved (RSVD)**
9:8	0h RW	**Command Completion Status (CCS):** 00 = Command not complete 01 = Command complete with Success 10 = Command complete with Error These bits must be cleared before a new command is initiated.
7:0	0h RO	**Read data (READ_DATA):** Data read as a result of the RRAP operation

16.1018 (PORT1_PROFILE_ATTRIBUTES_REG0)—Offset 890Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1019 (PORT1_PROFILE_ATTRIBUTES_REG1)—Offset 8910h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1020 (PORT1_PROFILE_ATTRIBUTES_REG2)—Offset 8914h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1021 (PORT1_PROFILE_ATTRIBUTES_REG3)—Offset 8918h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1022 (PORT1_PROFILE_ATTRIBUTES_REG4)—Offset 891Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1023 (PORT1_PROFILE_ATTRIBUTES_REG5)—Offset 8920h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1024 (PORT1_PROFILE_ATTRIBUTES_REG6)—Offset 8924h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1025 (PORT1_PROFILE_ATTRIBUTES_REG7)—Offset 8928h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1026 (PORT1_PROFILE_ATTRIBUTES_REG8)—Offset 892Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1027 (PORT1_PROFILE_ATTRIBUTES_REG9)—Offset 8930h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1028 (PORT1_PROFILE_ATTRIBUTES_REG10)—Offset 8934h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1029 (PORT1_PROFILE_ATTRIBUTES_REG11)—Offset 8938h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1030 (PORT1_PROFILE_ATTRIBUTES_REG12)—Offset 893Ch

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1031 (PORT1_PROFILE_ATTRIBUTES_REG13)—Offset 8940h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1032 (PORT1_PROFILE_ATTRIBUTES_REG14)—Offset 8944h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1033 (PORT1_PROFILE_ATTRIBUTES_REG15)—Offset 8948h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1034 (PORT1_PROFILE_ATTRIBUTES_REG16)—Offset 894Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1035 (PORT1_PROFILE_ATTRIBUTES_REG17)—Offset 8950h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1036 (PORT1_PROFILE_ATTRIBUTES_REG18)—Offset 8954h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1037 (PORT1_PROFILE_ATTRIBUTES_REG19)—Offset 8958h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1038 (PORT1_PROFILE_ATTRIBUTES_REG20)—Offset 895Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1039 (PORT1_PROFILE_ATTRIBUTES_REG21)—Offset 8960h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1040 (PORT1_PROFILE_ATTRIBUTES_REG22)—Offset 8964h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1041 (PORT1_PROFILE_ATTRIBUTES_REG23)—Offset 8968h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1042 (PORT1_PROFILE_ATTRIBUTES_REG24)—Offset 896Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1043 (PORT1_PROFILE_ATTRIBUTES_REG25)—Offset 8970h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1044 (PORT1_PROFILE_ATTRIBUTES_REG26)—Offset 8974h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1045 (PORT1_PROFILE_ATTRIBUTES_REG27)—Offset 8978h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1046 (PORT1_PROFILE_ATTRIBUTES_REG28)—Offset 897Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1047 (PORT1_PROFILE_ATTRIBUTES_REG29)—Offset 8980h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1048 (PORT1_PROFILE_ATTRIBUTES_REG30)—Offset 8984h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1049 (PORT1_PROFILE_ATTRIBUTES_REG31)—Offset 8988h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1050 (PORT1_PROFILE_ATTRIBUTES_REG32)—Offset 898Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1051 (PORT1_PROFILE_ATTRIBUTES_REG33)—Offset 8990h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1052 (PORT1_PROFILE_ATTRIBUTES_REG34)—Offset 8994h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1053 (PORT1_PROFILE_ATTRIBUTES_REG35)—Offset 8998h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1054 (PORT1_PROFILE_ATTRIBUTES_REG36)—Offset 899Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1055 (PORT1_PROFILE_ATTRIBUTES_REG37)—Offset 89A0h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1056 (PORT1_PROFILE_ATTRIBUTES_REG38)—Offset 89A4h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1057 (PORT1_PROFILE_ATTRIBUTES_REG39)—Offset 89A8h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1058 (PORT1_PROFILE_ATTRIBUTES_REG40)—Offset 89ACh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1059 (PORT1_PROFILE_ATTRIBUTES_REG41)—Offset 89B0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1060 (PORT1_PROFILE_ATTRIBUTES_REG42)—Offset 89B4h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1061 (PORT1_PROFILE_ATTRIBUTES_REG43)—Offset 89B8h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1062 (PORT1_PROFILE_ATTRIBUTES_REG44)—Offset 89BCh

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1063 (PORT1_PROFILE_ATTRIBUTES_REG45)—Offset 89C0h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1064 (PORT1_PROFILE_ATTRIBUTES_REG46)—Offset 89C4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1065 (PORT1_PROFILE_ATTRIBUTES_REG47)—Offset 89C8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1066 (PORT1_PROFILE_ATTRIBUTES_REG48)—Offset 89CCh

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1067 (PORT1_PROFILE_ATTRIBUTES_REG49)—Offset 89D0h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1068 (PORT1_PROFILE_ATTRIBUTES_REG50)—Offset 89D4h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1069 (PORT1_PROFILE_ATTRIBUTES_REG51)—Offset 89D8h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1070 (PORT1_PROFILE_ATTRIBUTES_REG52)—Offset 89DCh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1071 (PORT1_PROFILE_ATTRIBUTES_REG53)—Offset 89E0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1072 (PORT1_PROFILE_ATTRIBUTES_REG54)—Offset 89E4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1073 (PORT1_PROFILE_ATTRIBUTES_REG55)—Offset 89E8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1074 (PORT1_PROFILE_ATTRIBUTES_REG56)—Offset 89ECh

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1075 (PORT1_PROFILE_ATTRIBUTES_REG57)—Offset 89F0h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1076 (PORT1_PROFILE_ATTRIBUTES_REG58)—Offset 89F4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1077 (PORT1_PROFILE_ATTRIBUTES_REG59)—Offset 89F8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1078 (PORT1_PROFILE_ATTRIBUTES_REG60)—Offset 89FCh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1079 (PORT1_PROFILE_ATTRIBUTES_REG61)—Offset 8A00h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1080 (PORT1_PROFILE_ATTRIBUTES_REG62)—Offset 8A04h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1081 (PORT1_PROFILE_ATTRIBUTES_REG63)—Offset 8A08h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD)**
27:16	0h RW	**Attribute ID (ATT_ID)**
15	0h RW	**Valid (VALID):** When set, indicates that the attribute value in bits 7:0 must be written to the address specified in bits 27:16
14	0h RW	**TARGET_PHY (TARGET_PHY):** 0 = Remote Phy 1 = Local Phy
13:8	0h RO	**Reserved (RSVD1)**
7:0	0h RW	**ATTRIBUTE VALUE (ATT_VALUE)**

16.1082 GLOBAL_TIME_SYNC_CAP_REG (GLOBAL_TIME_SYNC_CAP_REG)—Offset 8E10h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 12C9h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED (RSVD)**
15:8	12h RO	**Next Capability pointer (NCP)**
7:0	C9h RO	**Capability ID (CID)**

16.1083 GLOBAL_TIME_SYNC_CTRL_REG (GLOBAL_TIME_SYNC_CTRL_REG)—Offset 8E14h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	**RESERVED (RSVD)**
0	0h RW/1S	**Time Stamp Counter Capture Initiate (TIME_STAMP_CNTR_CAPTURE_INITIATE):** SW sets this bit to initiate a time capture. Once the time capture is complete and the time values are valid in the Local and Global time capture registers, HW clears the bit.

16.1084 MICROFRAME_TIME_REG (MICROFRAME_TIME_REG)—Offset 8E18h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**RESERVED (RSVD)**
29:16	0h RO	**Captured Frame List Current Index/Frame Number (CMFI):** The value in this register is updated in response to sample_now signal. Bits [29:16] reflect state of bits [13:0] of FRINDEX
15:13	0h RO	**RESERVED1 (RSVD1)**
12:0	0h RO	**Captured Micro-frame BLIF (CMFB):** The value is updated in response to sample_now signal and provides information about offset within micro-frame. Captured value represents number of 8 high-speed bit time units from start of micro-frame. At the beginning of micro-frame captured value will be 0 and increase to maximum value at the end. Default maximum value is 7499 but it may be changed as result of adjustment done via Bus Interval Adjust (BIA).

16.1085 GLOBAL_TIME_LOW_REG (GLOBAL_TIME_LOW_REG)—Offset 8E20h

Global Time Value (Low):

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**GLOBAL_TIME_LOW_REG (GLOBAL_TIME_LOW):** Global Time Value (Low):

16.1086 GLOBAL_TIME_HI_REG (GLOBAL_TIME_HI_REG)—Offset 8E24h

Global Time Value (High):

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**GLOBAL_TIME_HI_REG (GLOBAL_TIME_HI):** Global Time Value (High):

16.1087 Debug Status Capability Register (DEBUG_STATUS_CAPABILITY_REG)—Offset 8E58h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: CBh

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Rsvd (Rsvd)**
15:8	0h RO	**Next Capability Pointer (NCP)**
7:0	CBh RO	**Capability ID (CID)**

16.1088 Host Ctrl USB3 Soft Error Count Register 1 (HOST_CTRL_USB3_ERR_COUNT_REG1)—Offset 8E5Ch

This register is updated by hardware and cleared by software

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED (cfg_reserved)**
15:0	0h RO	**USB3_SOFT_ERR_CNT (cfg_usb3_soft_error_cnt)**

16.1089 Host Ctrl USB3 Soft Error Count Register 2 (HOST_CTRL_USB3_ERR_COUNT_REG2)—Offset 8E60h

This register is updated by hardware and cleared by software

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED (cfg_reserved)**
15:0	0h RO	**USB3_SOFT_ERR_CNT (cfg_usb3_soft_error_cnt)**

16.1090 Host Ctrl USB3 Soft Error Count Register 3 (HOST_CTRL_USB3_ERR_COUNT_REG3)—Offset 8E64h

This register is updated by hardware and cleared by software

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED (cfg_reserved)**
15:0	0h RO	**USB3_SOFT_ERR_CNT (cfg_usb3_soft_error_cnt)**

16.1091 Host Ctrl USB3 Soft Error Count Register 4 (HOST_CTRL_USB3_ERR_COUNT_REG4)—Offset 8E68h

This register is updated by hardware and cleared by software

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED (cfg_reserved)**
15:0	0h RO	**USB3_SOFT_ERR_CNT (cfg_usb3_soft_error_cnt)**

16.1092 Host Ctrl USB3 Soft Error Count Register 5 (HOST_CTRL_USB3_ERR_COUNT_REG5)—Offset 8E6Ch

This register is updated by hardware and cleared by software

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED (cfg_reserved)**
15:0	0h RO	**USB3_SOFT_ERR_CNT (cfg_usb3_soft_error_cnt)**

16.1093 Host Ctrl USB3 Soft Error Count Register 6 (HOST_CTRL_USB3_ERR_COUNT_REG6)—Offset 8E70h

This register is updated by hardware and cleared by software

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED (cfg_reserved)**
15:0	0h RO	**USB3_SOFT_ERR_CNT (cfg_usb3_soft_error_cnt)**

16.1094 Host Ctrl USB3 Soft Error Count Register 7 (HOST_CTRL_USB3_ERR_COUNT_REG7)—Offset 8E74h

This register is updated by hardware and cleared by software

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED (cfg_reserved)**
15:0	0h RO	**USB3_SOFT_ERR_CNT (cfg_usb3_soft_error_cnt)**

16.1095 (GEN_REGRW1)—Offset B0h

General Purpose Read Write Register1

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW1**

16.1096 (GEN_REGRW2)—Offset B4h

General Purpose Read Write Register2

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW2**

16.1097 (GEN_REGRW3)—Offset B8h

General Purpose Read Write Register3

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW3**

16.1098 (GEN_REGRW4)—Offset BCh

General Purpose Read Write Register4

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RW	**GEN_REG_RW4:** reserved
2	0h RW	**XDCI_PIPE_CLK_GATE_DIS:** xdci pipe clock gating disable
1	0h RW	**XDCI_SB_CLK_GATE_DIS:** xdci sideband clock gating disable
0	0h RW	**XDCI_PRIM_CLK_GATE_DIS:** xdci primary clock gating disable

16.1099 (GEN_INPUT_REGRW)—Offset C0h

General Purpose Input Register

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 1008000Bh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**BVALID_HW (HW_BVALID):** hardware bvalid
30	0h RO	**gsts_buserraddvld:** Bus error
29	0h RO	**DRD_MODE (DRD_MODE):** SS DRD mode 0 device mode 1 host mode
28	1h RO	**IDPIN (IDPIN):** IDPIN value (debounce)
27:26	0h RO	**u2_dssr_state:** USB2 DSSR State

Bit Range	Default & Access	Field Name (ID): Description
25:21	0h RO	**u2_prt_state:** USB2 Port State
20:17	4h RO	**ltdb_link_state:** USB3 Link state
16:13	0h RO	**ltdb_sub_state:** USB3 Link sub state
12	0h RO	**reserved1**
11:10	0h RO	**current_power_state_u3pmu:** The current power state of the core. When equal to '3', the PMU is controlling the PHY, and when equal to '0', the core is controlling the PHY
9:8	0h RO	**current_power_state_u2pmu:** The current power state of the core. When equal to '3', the PMU is controlling the PHY, and when equal to '0', the core is controlling the PHY
7	0h RO	**connect_state_u3pmu:** When '1', indicates the PMU is maintaining at least one connection to the host or a device (the link is in U3). When '0', indicates the PMU has no connection to the host or any device.
6	0h RO	**connect_state_u2pmu:** When '1', indicates the PMU is maintaining at least one connection to the host or a device (the link is in L1, L2). When '0', indicates the PMU has no connection to the host or any device
5	0h RO	**utmi_l1_suspend_com_n:** Common L1 suspend
4	0h RO	**utmi_suspend_com_n:** Common suspend
3	1h RO	**utmi_suspend_n:** USB 2.0 Port Suspend
2:1	1h RO	**usb2_enumspeed:** Device Enumerated Speed
0	1h RO	**b2rl_cur_mode:** Current Mode. Current Mode 1'b0 Host ,1'b1 Device

16.1100 (GEN_REGRW1)—Offset B0h

General Purpose Read Write Register1

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 1

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW1**

16.1101 (GEN_REGRW2)—Offset B4h

General Purpose Read Write Register2

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 1

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW2**

16.1102 (GEN_REGRW3)—Offset B8h

General Purpose Read Write Register3

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 1

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW3**

16.1103 (GEN_REGRW4)—Offset BCh

General Purpose Read Write Register4

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 1

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RW	**GEN_REG_RW4:** reserved
2	0h RW	**XDCI_PIPE_CLK_GATE_DIS:** xdci pipe clock gating disable
1	0h RW	**XDCI_SB_CLK_GATE_DIS:** xdci sideband clock gating disable

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**XDCI_PRIM_CLK_GATE_DIS:** xdci primary clock gating disable

16.1104 (GEN_INPUT_REGRW)—Offset C0h

General Purpose Input Register

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 1

Default: 1008000Bh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**BVALID_HW (HW_BVALID):** hardware bvalid
30	0h RO	**gsts_buserraddvld:** Bus error
29	0h RO	**DRD_MODE (DRD_MODE):** SS DRD mode 0 device mode 1 host mode
28	1h RO	**IDPIN (IDPIN):** IDPIN value (debounce)
27:26	0h RO	**u2_dssr_state:** USB2 DSSR State
25:21	0h RO	**u2_prt_state:** USB2 Port State
20:17	4h RO	**ltdb_link_state:** USB3 Link state
16:13	0h RO	**ltdb_sub_state:** USB3 Link sub state
12	0h RO	**reserved1**
11:10	0h RO	**current_power_state_u3pmu:** The current power state of the core. When equal to '3', the PMU is controlling the PHY, and when equal to '0', the core is controlling the PHY
9:8	0h RO	**current_power_state_u2pmu:** The current power state of the core. When equal to '3', the PMU is controlling the PHY, and when equal to '0', the core is controlling the PHY
7	0h RO	**connect_state_u3pmu:** When '1', indicates the PMU is maintaining at least one connection to the host or a device (the link is in U3). When '0', indicates the PMU has no connection to the host or any device.
6	0h RO	**connect_state_u2pmu:** When '1', indicates the PMU is maintaining at least one connection to the host or a device (the link is in L1, L2). When '0', indicates the PMU has no connection to the host or any device

Bit Range	Default & Access	Field Name (ID): Description
5	0h RO	**utmi_l1_suspend_com_n:** Common L1 suspend
4	0h RO	**utmi_suspend_com_n:** Common suspend
3	1h RO	**utmi_suspend_n:** USB 2.0 Port Suspend
2:1	1h RO	**usb2_enumspeed:** Device Enumerated Speed
0	1h RO	**b2rl_cur_mode:** Current Mode. Current Mode 1'b0 Host ,1'b1 Device

16.1105 APBFC_U3PMU_CFG0 (APBFC_U3PMU_CFG0)—Offset 10F808h

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 800h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**DRD_CONTROL_BIT_15 (DRD_CONTROL_BIT_15)**
14:3	100h RW	**DEBOUNCE_VAL (DEBOUNCE_VAL):** ID ping debounce timer (DEBOUNCE_VAL): in the unit of RTC clock (33us) default to 8.448 ms
2	0h RW	**SYNCHRONIZE_SS_HS_SWITCH (SYNCHRONIZE_SS_HS_SWITCH):** Synchronize the SS and HS switch: 0 (default) Does not synchronize. i.e. HS switch on the debounced id pin, while SS switch independently controlled by the sequencer. 1 synchronize HS and SS switch. Both speeds switch when sequencer switch.
1:0	0h RW	**DRD_CONFIG (DRD_CONFIG):** 00 Dynamic DRD switch mode 01 static host mode 10 static device mode 11 -- reserved

16.1106 APBFC_U3PMU_CFG1 (APBFC_U3PMU_CFG1)—Offset 10F80Ch

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**XDCI_DRD_CTRL_EN (DRD_CONTROL_BIT_31):** When DRD_ACCESS_MODE (bit 2) =1, 0 -- The policies below is controlled by register in the XHCI MMIO 1 -- The policies below is controlled by register in the XDCI MMIO When DRD_ACCESS_MODE =0, this bit is a don?t care.
14:12	0h RW	**DRD_WDT_VAL (DRD_WDT_VAL):** DRD WDT timer: 0-based incremental of 500ms
11	0h RW	**EN_HOST_TO_DEV_WDT (EN_HOST_TO_DEV_WDT):** enable DRD WDT switch from host to device 0 - disable watchdog timer 1- enable watchdog timer
10	0h RW	**EN_DEV_TO_HOST_WDT (EN_DEV_TO_HOST_WDT):** enable DRD WDT switch from device to host 0 - disable watchdog timer 1- enable watchdog timer
9	0h RW	**EN_RST_WDT (EN_RST_WDT):** enable DRD WDT switch out of cold reset 0 - disable watchdog timer 1- enable watchdog timer
8	0h RW	**SW_VBUS_VALID (SW_VBUS_VALID):** if SW_IDPIN_EN (bit 5) is 1, 0 - deassert sw vbus valid 1 - assert sw vbus valid
7	0h RW	**RSVD1 (RSVD1):** RESERVED
6	0h RW	**EN_PIPE_RX_ON_IDPIN (EN_PIPE_RX_ON_IDPIN):** During the connection to a device, there may be a delay in DRD switch from XDCI to XHCI mode,the rx term can be low after idpin deasserts. If this bit is 0, the rx term will be assert immediately after idpin toggle. Otherwise, the device may fall back to USB2 mode. 1 -- drive 0s on utmi rx signals to controller if not connected.
5	0h RW	**SW_IDPIN_EN (SW_IDPIN_EN):** SW_IDPIN_EN 1 -- enable SW id pin (pre DRD) 0 -- disable SW id pin.
4	0h RW	**SW_IDPIN (SW_IDPIN):** if SW_IDPIN_EN (bit 21) is 1, 0 - host mode 1 - device mode
3	0h RW	**USB2_SUSP_OR_DIS (USB2_SUSP_OR_DIS):** 1 - The DRD UTMI suspendm will be controlled by host/device based on the DRD switch. 0 - whenever device or host deassert suspendm, the DRD UTMI suspendm will be deasserted.
2	0h RW	**DRD_ACCESS_MODE (DRD_ACCESS_MODE):** 0 - The DUAL_ROLE register from host and device are ORed to control DRD 1 - The DUAL_ROLE register from host and device are selected by XDCI_DRD_CTRL_EN (bit 31) to control DRD
1	0h RW	**EN_DIRECT_DRD_SWITCH_ON_USB3PORT_BY_IDPIN (EN_DIRECT_DRD_SWITCH_ON_USB3PORT_BY_IDPIN):** 1 enable the direct DRD switch on USB3 port by idpin 0 disable the direct DRD switch
0	0h RW	**SW_SWITCH_ENABLE (SW_SWITCH_ENABLE):** SW switch enable 0 (default) ID pin HW controlled DRD. 1 -- SW controlled DRD (ignore idpin), switch based on the DRD_CONFIG.

16.1107 APBFC_U3PMU_CFG2 (APBFC_U3PMU_CFG2)—Offset 10F810h

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: Bh

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	**ltdb_sub_state_2_0:** USB3 Link sub state[3:0]
12	0h RO	**reserved1**
11:10	0h RO	**current_power_state_u3pmu:** The current power state of the core. When equal to '3', the PMU is controlling the PHY, and when equal to '0', the core is controlling the PHY
9:8	0h RO	**current_power_state_u2pmu:** The current power state of the core. When equal to '3', the PMU is controlling the PHY, and when equal to '0', the core is controlling the PHY
7	0h RO	**connect_state_u3pmu:** When '1', indicates the PMU is maintaining at least one connection to the host or a device (the link is in U3). When '0', indicates the PMU has no connection to the host or any device.
6	0h RO	**connect_state_u2pmu:** When '1', indicates the PMU is maintaining at least one connection to the host or a device (the link is in L1, L2). When '0', indicates the PMU has no connection to the host or any device
5	0h RO	**utmi_l1_suspend_com_n:** Common L1 suspend
4	0h RO	**utmi_suspend_com_n:** Common suspend
3	1h RO	**utmi_suspend_n:** USB 2.0 Port Suspend
2:1	1h RO	**usb2_enumspeed:** Device Enumerated Speed
0	1h RO	**b2rl_cur_mode:** Current Mode. Current Mode 1'b0 Host ,1'b1 Device

16.1108 APBFC_U3PMU_CFG3 (APBFC_U3PMU_CFG3)—Offset 10F814h

General Purpose Input Register

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 1008h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**BVALID_HW (HW_BVALID):** hardware bvalid
14	0h RO	**gsts_buserraddvld:** Bus error
13	0h RO	**DRD_MODE (DRD_MODE):** SS DRD mode 0 device mode 1 host mode
12	1h RO	**IDPIN (IDPIN):** IDPIN value (debounce)
11:10	0h RO	**u2_dssr_state:** USB2 DSSR State
9:5	0h RO	**u2_prt_state:** USB2 Port State
4:1	4h RO	**ltdb_link_state:** USB3 Link state
0	0h RO	**ltdb_sub_state_3:** USB3 Link sub state[3]

16.1109 APBFC_U3PMU_CFG4 (APBFC_U3PMU_CFG4)—Offset 10F818h

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RW	**hub_port_perm_attach:** reserved
13	0h RW	**host_port_power_control:** reserved
12	0h RW	**xhci_revision:** reserved
11:8	0h RW	**bus_filter_bypass:** Bus Filter Bypass. Disables the internal bus filters that are enabled by DWC_USB3_EN_BUS_FILTERS coreConsultant parameter. This static signal is present only when DWC_USB3_EN_BUS_FILTERS is 1. It is expected that this signal is set or reset at power-on reset and is not changed during the normal operation of the core.
7	0h RW	**Reserved2:** reserved
6	0h RW	**Reserved1:** reserved

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**otg_phy_pwr_off_veto:** reserved
4	0h RW	**otg_cnt_pwr_off_veto:** Indicates that controller power should not power off even on RTD3hot.Also a veto signal to keep LPPLL on. Note: Currently LPPLL results in a veto for S0iX flows
3	0h RW	**Brdg_rst:** reset the core see also regRW1[31]
2	0h RW	**USB2PHY_BVALID_MASK_DIS:** 0: enable USB2 PHY BVALID mask 1: disable
1:0	0h RW	**pm_power_state_request:** This port defines the PCI power management state requested by the software. When the core is configured with two power rail support (DWC_USB3_EN_PWROPT=2), the valid states are: ? 00: D0 ? 11: D3 Active State

16.1110 APBFC_U3PMU_CFG5 (APBFC_U3PMU_CFG5)—Offset 10F81Ch

Access Method

Type: MEM Register (Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**Brdg_rst_mux:** This is a mux the core reset between the normal reset or reset and with bdg_reset regRW1[3]
14	0h RW	**En_otg_interrupt:** reserved
13:8	0h RW	**fladj_30mhz_reg:** HS jitter adjusment
7	0h RW	**core_hub_port_overcurrent:** This is a generic register provided for product specific behavior
6	0h RW	**GEN_REG_RW1:** This is a generic register provided for product specific behavior
5	0h RW	**otg_phy_pwr_off_req:** reserved
4	0h RW	**u2_pme_en:** Determines whether ULPI flis PME events are allowed to trigger PME events to brige/GPIO
3	0h RW	**u3_pme_en:** Determines whether USB3 flis PME events are allowed to trigger PME events to brige/GPIO
2	0h RW	**core_pme_en:** Determines whether core PME events are allowed to trigger PME events to brige/GPIO

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**ulpiphy_refclk_disable:** reserved
0	0h RW	**ipma_cmn_refclk_disable:** reserved

16.1111 APBFC_U3PMU_CFG6 (APBFC_U3PMU_CFG6)—Offset 10F820h

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**HW_BVALID_DEV:** HW Vbus Bvalid indication
14	0h RO	**RESERVED (RSVD2)**
13	0h RO	**HOST_OWNED_DEV:** DRD mode,0 is device mode 1 is host mode
12	1h RO	**IDPIN_DEV:** ID pin Debounced version
11:0	0h RO	**RESERVED (RSVD3)**

16.1112 APBFC_D0I3C (APBFC_D0I3C)—Offset 10F830h

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 8h

Bit Range	Default & Access	Field Name (ID): Description
15:4	0h RW	**RESERVED (RSVD)**
3	1h RW/1C	**RestoreRequired (RESTORE_REQUIRED):** When set (by HW), SW must restore state to the IP. The state may have been lost due to a reset or full power lost. SW clears the bit by writing a 1. This bit will be set on initial power up.
2	0h RW	**D0I3 (D0I3):** SW sets this bit to 1 to move the IP into the D0i3 state. Writing this bit to 0 will return the IP to the fully active D0 state (D0i0).

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**RESERVED (RSVD1)**

GSBUSCFG0 (GSBUSCFG0)—Offset C100h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 6h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**reserved_32_16 (reserved_32_16)**
15	0h RW	**reserved_16_15 (reserved_16_15)**
14:13	0h RW	**reserved_15_13 (reserved_15_13)**
12	0h RW	**STOREANDFORWARD (STOREANDFORWARD)**
11	0h RW	**DATBIGEND (DATBIGEND)**
10	0h RW	**DESBIGEND (DESBIGEND)**
9:8	0h RO	**reserved_10_8 (reserved_10_8)**
7	0h RW	**INCR256BRSTENA (INCR256BRSTENA)**
6	0h RW	**INCR128BRSTENA (INCR128BRSTENA)**
5	0h RW	**INCR64BRSTENA (INCR64BRSTENA)**
4	0h RW	**INCR32BRSTENA (INCR32BRSTENA)**
3	0h RW	**INCR16BRSTENA (INCR16BRSTENA)**
2	1h RW	**INCR8BRSTENA (INCR8BRSTENA)**
1	1h RW	**INCR4BRSTENA (INCR4BRSTENA)**
0	0h RW	**INCRBRSTENA (INCRBRSTENA)**

GSBUSCFG1 (GSBUSCFG1)—Offset C104h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: F00h

Bit Range	Default & Access	Field Name (ID): Description
31:13	0h RO	**reserved_32_13 (reserved_32_13)**
12	0h RW	**EN1KPAGE (EN1KPAGE)**
11:8	Fh RW	**PipeTransLimit (PipeTransLimit)**
7:4	0h RW	**DATADRSPC (DATADRSPC)**
3:0	0h RW	**DESADRSPC (DESADRSPC)**

16.1113 GTXTHRCFG (GTXTHRCFG)—Offset C108h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**USBISOTHREN (USBISOTHREN)**
30	0h RW	**reserved_31_30 (reserved_31_30)**
29	0h RO	**USBTxPktCntSel (USBTxPktCntSel)**
28	0h RO	**reserved_29_28 (reserved_29_28)**
27:24	0h RO	**USBTxPktCnt (USBTxPktCnt)**
23:16	0h RW	**USBMaxTxBurstSize (USBMaxTxBurstSize)**
15	0h RW	**Reserved_15_15 (Reserved_15_15)**
14	0h RW	**reserved_15_14 (reserved_15_14)**
13:11	0h RO	**reserved_14_11 (reserved_14_11)**
10:0	0h RW	**reserved_11_0 (reserved_11_0)**

16.1114 GRXTHRCFG (GRXTHRCFG)—Offset C10Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 24400000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**reserved_32_30 (reserved_32_30)**
29	1h RW	**USBRxPktCntSel (USBRxPktCntSel)**
28	0h RO	**reserved_29_28 (reserved_29_28)**
27:24	4h RW	**USBRxPktCnt (USBRxPktCnt)**
23:19	8h RW	**USBMaxRxBurstSize (USBMaxRxBurstSize)**
18:16	0h RO	**reserved_19_16 (reserved_19_16)**
15	0h RW	**reserved_16_15 (reserved_16_15)**
14:11	0h RO	**reserved_15_11 (reserved_15_11)**
10:0	0h RO	**reserved_11_0 (reserved_11_0)**

16.1115 GCTL (GCTL)—Offset C110h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 102000h

Bit Range	Default & Access	Field Name (ID): Description
31:19	2h RW	**PWRDNSCALE (PWRDNSCALE)**
18	0h RW	**MASTERFILTBYPASS (MASTERFILTBYPASS)**
17	0h RW	**BYPSSETADDR (BYPSSETADDR)**

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW	**U2RSTECN (U2RSTECN)**
15:14	0h RW	**FRMSCLDWN (FRMSCLDWN)**
13:12	2h RW	**PRTCAPDIR (PRTCAPDIR)**
11	0h RW	**CORESOFTRESET (CORESOFTRESET)**
10	0h RW	**SOFITPSYNC (SOFITPSYNC)**
9	0h RW	**U1U2TimerScale (U1U2TimerScale)**
8	0h RW	**DEBUGATTACH (DEBUGATTACH)**
7:6	0h RW	**RAMCLKSEL (RAMCLKSEL)**
5:4	0h RW	**SCALEDOWN (SCALEDOWN)**
3	0h RW	**DISSCRAMBLE (DISSCRAMBLE)**
2	0h RW	**U2EXIT_LFPS (U2EXIT_LFPS)**
1	0h RW	**GblHibernationEn (GblHibernationEn)**
0	0h RW	**DSBLCLKGTNG (DSBLCLKGTNG)**

16.1116 GPMSTS (GPMSTS)—Offset C114h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h WO	**PortSel (PortSel)**
27:16	0h RW	**Reserved_16_27 (Reserved_16_27)**
15:12	0h RO	**U3Wakeup (U3Wakeup)**
11:9	0h RW	**Reserved_9_11 (Reserved_9_11)**

Bit Range	Default & Access	Field Name (ID): Description
8:0	0h RO	**U2Wakeup (U2Wakeup)**

16.1117 GSTS (GSTS)—Offset C118h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 3E800000h

Bit Range	Default & Access	Field Name (ID): Description
31:20	3E8h RO	**CBELT (CBELT)**
19:12	0h RO	**reserved_20_12 (reserved_20_12)**
11	0h RO	**SSIC_IP (SSIC_IP)**
10	0h RO	**OTG_IP (OTG_IP)**
9	0h RO	**BC_IP (BC_IP)**
8	0h RO	**ADP_IP (ADP_IP)**
7	0h RO	**Host_IP (Host_IP)**
6	0h RO	**Device_IP (Device_IP)**
5	0h RO	**CSRTimeout (CSRTimeout)**
4	0h RO	**BUSERRADDRVLD (BUSERRADDRVLD)**
3:2	0h RO	**reserved_4_2 (reserved_4_2)**
1:0	0h RO	**CURMOD (CURMOD)**

16.1118 GUCTL1 (GUCTL1)—Offset C11Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RW	**reserved_32_15 (reserved_32_15)**
14:9	0h RW	**Reserved_15_9 (Reserved_15_9)**
8	0h RW	**L1_SUSP_THRLD_EN_FOR_HOST (L1_SUSP_THRLD_EN_FOR_HOST)**
7:4	0h RW	**L1_SUSP_THRLD_FOR_HOST (L1_SUSP_THRLD_FOR_HOST)**
3	0h RW	**HC_ERRATA_ENABLE (HC_ERRATA_ENABLE)**
2	0h RW	**HC_PARCHK_DISABLE (HC_PARCHK_DISABLE)**
1	1h RW	**OVRLD_L1_SUSP_COM (OVRLD_L1_SUSP_COM)**
0	0h RW	**LOA_FILTER_EN (LOA_FILTER_EN)**

16.1119 GSNPSID (GSNPSID)—Offset C120h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 5533260Ah

Bit Range	Default & Access	Field Name (ID): Description
31:0	5533260Ah RO	**SYNOPSYSID (SYNOPSYSID)**

16.1120 GGPIO (GGPIO)—Offset C124h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**GPO (GPO)**
15:0	0h RO	**GPI (GPI)**

16.1121 GUID (GUID)—Offset C128h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 8086A0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	8086A0h RW	**USERID (USERID)**

16.1122 GUCTL (GUCTL)—Offset C12Ch

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 10h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RW	**REFCLKPER (REFCLKPER)**
21	0h RW	**NoExtrDl (NoExtrDl)**
20:18	0h RW	**PSQExtrResSp (PSQExtrResSp)**
17	0h RW	**SprsCtrlTransEn (SprsCtrlTransEn)**
16	0h RO	**reserved_17_16 (reserved_17_16)**
15	0h RO	**reserved_16_15 (reserved_16_15)**
14	0h RW	**USBHstInAutoRetryEn (USBHstInAutoRetryEn)**

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**EnOverlapChk (EnOverlapChk)**
12	0h RW	**ExtCapSupptEN (ExtCapSupptEN)**
11	0h RW	**InsrtExtrFSBODI (InsrtExtrFSBODI)**
10:9	0h RW	**DTCT (DTCT)**
8:0	10h RW	**DTFT (DTFT)**

16.1123 GBUSERRADDRLO (GBUSERRADDRLO)—Offset C130h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**BUSERRADDR (BUSERRADDR)**

16.1124 GBUSERRADDRHI (GBUSERRADDRHI)—Offset C134h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**BUSERRADDR (BUSERRADDR)**

16.1125 GPRTBIMAPLO (GPRTBIMAPLO)—Offset C138h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**BINUM8 (BINUM8)**
27:24	0h RO	**BINUM7 (BINUM7)**
23:20	0h RO	**BINUM6 (BINUM6)**
19:16	0h RO	**BINUM5 (BINUM5)**
15:12	0h RO	**BINUM4 (BINUM4)**
11:8	0h RO	**BINUM3 (BINUM3)**
7:4	0h RO	**BINUM2 (BINUM2)**
3:0	0h RW	**BINUM1 (BINUM1)**

16.1126 GPRTBIMAPHI (GPRTBIMAPHI)—Offset C13Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**reserved_32_28 (reserved_32_28)**
27:24	0h RO	**BINUM15 (BINUM15)**
23:20	0h RO	**BINUM14 (BINUM14)**
19:16	0h RO	**BINUM13 (BINUM13)**
15:12	0h RO	**BINUM12 (BINUM12)**
11:8	0h RO	**BINUM11 (BINUM11)**
7:4	0h RO	**BINUM10 (BINUM10)**

Bit Range	Default & Access	Field Name (ID): Description
3:0	0h RO	**BINUM9 (BINUM9)**

16.1127 GHWPARAMS0 (GHWPARAMS0)—Offset C140h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 40204008h

Bit Range	Default & Access	Field Name (ID): Description
31:24	40h RO	**DWC_USB3_ADWIDTH_31_24 (DWC_USB3_ADWIDTH_31_24)**
23:16	20h RO	**DWC_USB3_SDWIDTH_23_16 (DWC_USB3_SDWIDTH_23_16)**
15:8	40h RO	**DWC_USB3_MDWIDTH_15_8 (DWC_USB3_MDWIDTH_15_8)**
7:6	0h RO	**DWC_USB3_SBUS_TYPE_7_6 (DWC_USB3_SBUS_TYPE_7_6)**
5:3	1h RO	**DWC_USB3_MBUS_TYPE_5_3 (DWC_USB3_MBUS_TYPE_5_3)**
2:0	0h RO	**DWC_USB3_MODE_2_0 (DWC_USB3_MODE_2_0)**

16.1128 GHWPARAMS1 (GHWPARAMS1)—Offset C144h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 260C93Bh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved_31 (Reserved_31)**
30	0h RO	**DWC_USB3_RM_OPT_FEATURES_30 (DWC_USB3_RM_OPT_FEATURES_30)**
29	0h RO	**Reserved1_29 (Reserved1_29)**

Bit Range	Default & Access	Field Name (ID): Description
28	0h RO	**DWC_USB3_RAM_BUS_CLKS_SYNC_28 (DWC_USB3_RAM_BUS_CLKS_SYNC_28)**
27	0h RO	**DWC_USB3_MAC_RAM_CLKS_SYNC_27 (DWC_USB3_MAC_RAM_CLKS_SYNC_27)**
26	0h RO	**DWC_USB3_MAC_PHY_CLKS_SYNC_26 (DWC_USB3_MAC_PHY_CLKS_SYNC_26)**
25:24	2h RO	**DWC_USB3_EN_PWROPT_25_24 (DWC_USB3_EN_PWROPT_25_24)**
23	0h RO	**DWC_USB3_SPRAM_TYP_23 (DWC_USB3_SPRAM_TYP_23)**
22:21	3h RO	**DWC_USB3_NUM_RAMS_22_21 (DWC_USB3_NUM_RAMS_22_21)**
20:15	1h RO	**DWC_USB3_DEVICE_NUM_INT_20_15 (DWC_USB3_DEVICE_NUM_INT_20_15)**
14:12	4h RO	**DWC_USB3_ASPACEWIDTH_14_12 (DWC_USB3_ASPACEWIDTH_14_12)**
11:9	4h RO	**DWC_USB3_REQINFOWIDTH_11_9 (DWC_USB3_REQINFOWIDTH_11_9)**
8:6	4h RO	**DWC_USB3_DATAINFOWIDTH_8_6 (DWC_USB3_DATAINFOWIDTH_8_6)**
5:3	7h RO	**DWC_USB3_BURSTWIDTH_5_3 (DWC_USB3_BURSTWIDTH_5_3)**
2:0	3h RO	**DWC_USB3_IDWIDTH_2_0 (DWC_USB3_IDWIDTH_2_0)**

16.1129 GHWPARAMS2 (GHWPARAMS2)—Offset C148h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 8086A0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	8086A0h RO	**DWC_USB3_USERID_31_0 (DWC_USB3_USERID_31_0)**

16.1130 GHWPARAMS3 (GHWPARAMS3)—Offset C14Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 10420085h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved_31 (Reserved_31)**
30:23	20h RO	**DWC_USB3_CACHE_TOTAL_XFER_RESOURCES_30_23 (DWC_USB3_CACHE_TOTAL_XFER_RESOURCES_30_23)**
22:18	10h RO	**DWC_USB3_NUM_IN_EPS_22_18 (DWC_USB3_NUM_IN_EPS_22_18)**
17:12	20h RO	**DWC_USB3_NUM_EPS_17_12 (DWC_USB3_NUM_EPS_17_12)**
11	0h RO	**DWC_USB3_ULPI_CARKIT_11 (DWC_USB3_ULPI_CARKIT_11)**
10	0h RO	**DWC_USB3_VENDOR_CTL_INTERFACE_10 (DWC_USB3_VENDOR_CTL_INTERFACE_10)**
9:8	0h RO	**ghwparams3_9_8 (ghwparams3_9_8)**
7:6	2h RO	**DWC_USB3_HSPHY_DWIDTH_7_6 (DWC_USB3_HSPHY_DWIDTH_7_6)**
5:4	0h RO	**DWC_USB3_FSPHY_INTERFACE_5_4 (DWC_USB3_FSPHY_INTERFACE_5_4)**
3:2	1h RO	**DWC_USB3_HSPHY_INTERFACE_3_2 (DWC_USB3_HSPHY_INTERFACE_3_2)**
1:0	1h RO	**DWC_USB3_SSPHY_INTERFACE_1_0 (DWC_USB3_SSPHY_INTERFACE_1_0)**

16.1131 GHWPARAMS4 (GHWPARAMS4)—Offset C150h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 47A22004h

Bit Range	Default & Access	Field Name (ID): Description
31:28	4h RO	**DWC_USB3_BMU_LSP_DEPTH_31_28 (DWC_USB3_BMU_LSP_DEPTH_31_28)**
27:24	7h RO	**DWC_USB3_BMU_PTL_DEPTH_27_24 (DWC_USB3_BMU_PTL_DEPTH_27_24)**

Bit Range	Default & Access	Field Name (ID): Description
23	1h RO	**DWC_USB3_EN_ISOC_SUPT_23 (DWC_USB3_EN_ISOC_SUPT_23)**
22	0h RO	**Reserved_22 (Reserved_22)**
21	1h RO	**DWC_USB3_EXT_BUFF_CONTROL_21 (DWC_USB3_EXT_BUFF_CONTROL_21)**
20:17	1h RO	**DWC_USB3_NUM_SS_USB_INSTANCES_20_17 (DWC_USB3_NUM_SS_USB_INSTANCES_20_17)**
16:13	1h RO	**DWC_USB3_HIBER_SCRATCHBUFS_16_13 (DWC_USB3_HIBER_SCRATCHBUFS_16_13)**
12	0h RO	**ghwparams4_12 (ghwparams4_12)**
11	0h RO	**ghwparams4_11 (ghwparams4_11)**
10:9	0h RO	**ghwparams4_10_9 (ghwparams4_10_9)**
8:7	0h RO	**ghwparams4_8_7 (ghwparams4_8_7)**
6	0h RO	**ghwparams4_6 (ghwparams4_6)**
5:0	4h RO	**DWC_USB3_CACHE_TRBS_PER_TRANSFER_5_0 (DWC_USB3_CACHE_TRBS_PER_TRANSFER_5_0)**

16.1132 GHWPARAMS5 (GHWPARAMS5)—Offset C154h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 4202088h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved_31_28 (Reserved_31_28)**
27:22	10h RO	**DWC_USB3_DFQ_FIFO_DEPTH_27_22 (DWC_USB3_DFQ_FIFO_DEPTH_27_22)**
21:16	20h RO	**DWC_USB3_DWQ_FIFO_DEPTH_21_16 (DWC_USB3_DWQ_FIFO_DEPTH_21_16)**
15:10	8h RO	**DWC_USB3_TXQ_FIFO_DEPTH_15_10 (DWC_USB3_TXQ_FIFO_DEPTH_15_10)**
9:4	8h RO	**DWC_USB3_RXQ_FIFO_DEPTH_9_4 (DWC_USB3_RXQ_FIFO_DEPTH_9_4)**

Bit Range	Default & Access	Field Name (ID): Description
3:0	8h RO	**DWC_USB3_BMU_BUSGM_DEPTH_3_0 (DWC_USB3_BMU_BUSGM_DEPTH_3_0)**

16.1133 GHWPARAMS6 (GHWPARAMS6)—Offset C158h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 2F60020h

Bit Range	Default & Access	Field Name (ID): Description
31:16	2F6h RO	**DWC_USB3_RAM0_DEPTH_31_16 (DWC_USB3_RAM0_DEPTH_31_16)**
15	0h RO	**BusFltrsSupport (BusFltrsSupport)**
14	0h RO	**BCSupport (BCSupport)**
13	0h RO	**OTG_SS_Support (OTG_SS_Support)**
12	0h RO	**ADPSupport (ADPSupport)**
11	0h RO	**HNPSupport (HNPSupport)**
10	0h RO	**SRPSupport (SRPSupport)**
9:8	0h RO	**Reserved_9_8 (Reserved_9_8)**
7	0h RO	**DWC_USB3_EN_FPGA_7 (DWC_USB3_EN_FPGA_7)**
6	0h RO	**Reserved1_6 (Reserved1_6)**
5:0	20h RO	**DWC_USB3_PSQ_FIFO_DEPTH_5_0 (DWC_USB3_PSQ_FIFO_DEPTH_5_0)**

16.1134 GHWPARAMS7 (GHWPARAMS7)—Offset C15Ch

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 38507E6h

Bit Range	Default & Access	Field Name (ID): Description
31:16	385h RO	**DWC_USB3_RAM2_DEPTH_31_16 (DWC_USB3_RAM2_DEPTH_31_16)**
15:0	7E6h RO	**DWC_USB3_RAM1_DEPTH_15_0 (DWC_USB3_RAM1_DEPTH_15_0)**

16.1135 GDBGFIFOSPACE (GDBGFIFOSPACE)—Offset C160h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 420000h

Bit Range	Default & Access	Field Name (ID): Description
31:16	42h RO	**SPACE_AVAILABLE (SPACE_AVAILABLE)**
15:8	0h RO	**reserved_16_8 (reserved_16_8)**
7:0	0h RW	**FIFO_QUEUE_SELECT (FIFO_QUEUE_SELECT)**

16.1136 GDBGLTSSM (GDBGLTSSM)—Offset C164h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 1010440h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**reserved_32_30 (reserved_32_30)**
29	0h RO	**X3_XS_SWAPPING (X3_XS_SWAPPING)**
28	0h RO	**X3_DS_HOST_SHUTDOWN (X3_DS_HOST_SHUTDOWN)**
27	0h RO	**PRTDIRECTION (PRTDIRECTION)**
26	0h RO	**LTDBTIMEOUT (LTDBTIMEOUT)**

Bit Range	Default & Access	Field Name (ID): Description
25:22	4h RO	**LTDBLINKSTATE (LTDBLINKSTATE)**
21:18	0h RO	**LTDBSUBSTATE (LTDBSUBSTATE)**
17	0h RO	**ELASTICBUFFERMODE (ELASTICBUFFERMODE)**
16	1h RO	**TXELECLDLE (TXELECLDLE)**
15	0h RO	**RXPOLARITY (RXPOLARITY)**
14	0h RO	**TxDetRxLoopback (TxDetRxLoopback)**
13:11	0h RO	**LTDBPhyCmdState (LTDBPhyCmdState)**
10:9	2h RO	**POWERDOWN (POWERDOWN)**
8	0h RO	**RXEQTRAIN (RXEQTRAIN)**
7:6	1h RO	**TXDEEMPHASIS (TXDEEMPHASIS)**
5:3	0h RO	**LTDBClkState (LTDBClkState)**
2	0h RO	**TXSWING (TXSWING)**
1	0h RO	**RXTERMINATION (RXTERMINATION)**
0	0h RO	**TXONESZEROS (TXONESZEROS)**

16.1137 GDBGLNMCC (GDBGLNMCC)—Offset C168h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Reserved_31_31 (Reserved_31_31)**
30:9	0h RO	**reserved_31_9 (reserved_31_9)**
8:0	0h RO	**LNMCC_BERC (LNMCC_BERC)**

16.1138 GDBGBMU (GDBGBMU)—Offset C16Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**BMU_BCU (BMU_BCU)**
7:4	0h RO	**BMU_DCU (BMU_DCU)**
3:0	0h RO	**BMU_CCU (BMU_CCU)**

16.1139 GDBGLSPMUX_DEV (GDBGLSPMUX_DEV)—Offset C170h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 3F0000h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**reserved_32_24 (reserved_32_24)**
23:16	3Fh RW	**logic_analyzer_trace (logic_analyzer_trace)**
15:14	0h RW	**reserved_16_14 (reserved_16_14)**
13:8	0h RW	**HOSTSELECT (HOSTSELECT)**
7:4	0h RW	**DEVSELECT (DEVSELECT)**
3:0	0h RW	**EPSELECT (EPSELECT)**

16.1140 GDBGLSP (GDBGLSP)—Offset C174h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**LSPDEBUG (LSPDEBUG)**

16.1141 GDBGEPINFO0 (GDBGEPINFO0)—Offset C178h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**EPDEBUG (EPDEBUG)**

16.1142 GDBGEPINFO1 (GDBGEPINFO1)—Offset C17Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 800000h

Bit Range	Default & Access	Field Name (ID): Description
31:0	800000h RO	**EPDEBUG (EPDEBUG)**

16.1143 GPRTBIMAP_HSLO (GPRTBIMAP_HSLO)—Offset C180h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**BINUM8 (BINUM8)**
27:24	0h RO	**BINUM7 (BINUM7)**
23:20	0h RO	**BINUM6 (BINUM6)**
19:16	0h RO	**BINUM5 (BINUM5)**
15:12	0h RO	**BINUM4 (BINUM4)**
11:8	0h RO	**BINUM3 (BINUM3)**
7:4	0h RO	**BINUM2 (BINUM2)**
3:0	0h RW	**BINUM1 (BINUM1)**

16.1144 GPRTBIMAP_HSHI (GPRTBIMAP_HSHI)—Offset C184h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**reserved_32_28 (reserved_32_28)**
27:24	0h RO	**BINUM15 (BINUM15)**
23:20	0h RO	**BINUM14 (BINUM14)**
19:16	0h RO	**BINUM13 (BINUM13)**
15:12	0h RO	**BINUM12 (BINUM12)**
11:8	0h RO	**BINUM11 (BINUM11)**
7:4	0h RO	**BINUM10 (BINUM10)**
3:0	0h RO	**BINUM9 (BINUM9)**

16.1145 GPRTBIMAP_FSLO (GPRTBIMAP_FSLO)—Offset C188h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**BINUM8 (BINUM8)**
27:24	0h RO	**BINUM7 (BINUM7)**
23:20	0h RO	**BINUM6 (BINUM6)**
19:16	0h RO	**BINUM5 (BINUM5)**
15:12	0h RO	**BINUM4 (BINUM4)**
11:8	0h RO	**BINUM3 (BINUM3)**
7:4	0h RO	**BINUM2 (BINUM2)**
3:0	0h RW	**BINUM1 (BINUM1)**

16.1146 GPRTBIMAP_FSHI (GPRTBIMAP_FSHI)—Offset C18Ch

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**reserved_32_28 (reserved_32_28)**
27:24	0h RO	**BINUM15 (BINUM15)**
23:20	0h RO	**BINUM14 (BINUM14)**
19:16	0h RO	**BINUM13 (BINUM13)**

Bit Range	Default & Access	Field Name (ID): Description
15:12	0h RO	**BINUM12 (BINUM12)**
11:8	0h RO	**BINUM11 (BINUM11)**
7:4	0h RO	**BINUM10 (BINUM10)**
3:0	0h RO	**BINUM9 (BINUM9)**

16.1147 Reserved_94 (Reserved_94)—Offset C194h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**reserved_32_0 (reserved_32_0)**

16.1148 Reserved_98 (Reserved_98)—Offset C198h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**reserved_32_0 (reserved_32_0)**

16.1149 GUSB2PHYCFG_0 (GUSB2PHYCFG_0)—Offset C200h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2400h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**PHYSOFTRST (PHYSOFTRST)**
30	0h RW	**U2_FREECLK_EXISTS (U2_FREECLK_EXISTS)**
29	0h RW	**ULPI_LPM_WITH_OPMODE_CHK (ULPI_LPM_WITH_OPMODE_CHK)**
28:27	0h RW	**HSIC_CON_WIDTH_ADJ (HSIC_CON_WIDTH_ADJ)**
26	0h RW	**INV_SEL_HSIC (INV_SEL_HSIC)**
25:19	0h RO	**reserved_25_19 (reserved_25_19)**
18	0h RW	**ULPIEXTVBUSINDIACTOR (ULPIEXTVBUSINDIACTOR)**
17	0h RW	**ULPIEXTVBUSDRV (ULPIEXTVBUSDRV)**
16	0h RW	**ULPICLKSUSM (ULPICLKSUSM)**
15	0h RW	**ULPIAUTORES (ULPIAUTORES)**
14	0h RO	**reserved_15_14 (reserved_15_14)**
13:10	9h RW	**USBTRDTIM (USBTRDTIM)**
9	0h RW	**XCVRDLY (XCVRDLY)**
8	0h RW	**ENBLSLPM (ENBLSLPM)**
7	0h WO	**PHYSEL (PHYSEL)**
6	0h RW	**SUSPENDUSB20 (SUSPENDUSB20)**
5	0h RO	**FSINTF (FSINTF)**
4	0h RO	**ULPI_UTMI_Sel (ULPI_UTMI_Sel)**
3	0h RW	**PHYIF (PHYIF)**
2:0	0h RW	**B1L (B1L)**

16.1150 GUSB2I2CCTL_0 (GUSB2I2CCTL_0)—Offset C240h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**reserved_32_0 (reserved_32_0)**

16.1151 GUSB2PHYACC_ULPI_0 (GUSB2PHYACC_ULPI_0)—Offset C280h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:27	0h RO	**reserved_32_27 (reserved_32_27)**
26	0h RO	**DISUIPIDRVR (DISUIPIDRVR)**
25	0h RO	**NEWREGREQ (NEWREGREQ)**
24	0h RO	**VSTSDONE (VSTSDONE)**
23	0h RO	**VSTSBSY (VSTSBSY)**
22	0h RO	**REGWR (REGWR)**
21:16	0h RO	**REGADDR (REGADDR)**
15:13	0h RW	**Reserved_13_15 (Reserved_13_15)**
12:8	0h RO	**EXTREGADDR (EXTREGADDR)**
7:0	0h RO	**REGDATA (REGDATA)**

16.1152 GUSB3PIPECTL_0 (GUSB3PIPECTL_0)—Offset C2C0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2020002h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**PHYSoftRst (PHYSoftRst)**
30	0h RW	**HstPrtCmpl (HstPrtCmpl)**
29	0h RW	**U2SSInactP3ok (U2SSInactP3ok)**
28	0h RW	**DisRxDetP3 (DisRxDetP3)**
27	0h RW	**Ux_exit_in_Px (Ux_exit_in_Px)**
26	0h RW	**ping_enhancement_en (ping_enhancement_en)**
25	1h RW	**u1u2exitfail_to_recov (u1u2exitfail_to_recov)**
24	0h RW	**request_p1p2p3 (request_p1p2p3)**
23	0h RW	**StartRxDetU3RxDet (StartRxDetU3RxDet)**
22	0h RW	**DisRxDetU3RxDet (DisRxDetU3RxDet)**
21:19	0h RW	**DelayP1P2P3 (DelayP1P2P3)**
18	0h RW	**DELAYP1TRANS (DELAYP1TRANS)**
17	1h RW	**SUSPENDENABLE (SUSPENDENABLE)**
16:15	0h RO	**DATWIDTH (DATWIDTH)**
14	0h RW	**AbortRxDetInU2 (AbortRxDetInU2)**
13	0h RW	**SkipRxDet (SkipRxDet)**
12	0h RW	**LFPSP0Algn (LFPSP0Algn)**
11	0h RW	**P3P2TranOK (P3P2TranOK)**
10	0h RW	**P3ExSigP2 (P3ExSigP2)**
9	0h RW	**LFPSFILTER (LFPSFILTER)**
8	0h RW	**RX_DETECT_to_Polling_LFPS_Control (RX_DETECT_to_Polling_LFPS_Control)**

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**SSICEn (SSICEn)**
6	0h RW	**TX_SWING (TX_SWING)**
5:3	0h RW	**TX_MARGIN (TX_MARGIN)**
2:1	1h RW	**TX_DE_EPPHASIS (TX_DE_EPPHASIS)**
0	0h RW	**ELASTIC_BUFFER_MODE (ELASTIC_BUFFER_MODE)**

16.1153 GTXFIFOSIZ0_0 (GTXFIFOSIZ0_0)—Offset C300h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 42h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**TXFSTADDR_N (TXFSTADDR_N)**
15:0	42h RW	**TXFDEP_N (TXFDEP_N)**

16.1154 GTXFIFOSIZ1_0 (GTXFIFOSIZ1_0)—Offset C304h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 420184h

Bit Range	Default & Access	Field Name (ID): Description
31:16	42h RW	**TXFSTADDR_N (TXFSTADDR_N)**
15:0	184h RW	**TXFDEP_N (TXFDEP_N)**

16.1155 GTXFIFOSIZ2_0 (GTXFIFOSIZ2_0)—Offset C308h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 1C60184h

Bit Range	Default & Access	Field Name (ID): Description
31:16	1C6h RW	**TXFSTADDR_N (TXFSTADDR_N)**
15:0	184h RW	**TXFDEP_N (TXFDEP_N)**

16.1156 GTXFIFOSIZ3_0 (GTXFIFOSIZ3_0)—Offset C30Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 34A0184h

Bit Range	Default & Access	Field Name (ID): Description
31:16	34Ah RW	**TXFSTADDR_N (TXFSTADDR_N)**
15:0	184h RW	**TXFDEP_N (TXFDEP_N)**

16.1157 GTXFIFOSIZ4_0 (GTXFIFOSIZ4_0)—Offset C310h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 4CE0082h

Bit Range	Default & Access	Field Name (ID): Description
31:16	4CEh RW	**TXFSTADDR_N (TXFSTADDR_N)**

Bit Range	Default & Access	Field Name (ID): Description
15:0	82h RW	**TXFDEP_N (TXFDEP_N)**

16.1158 GTXFIFOSIZ5_0 (GTXFIFOSIZ5_0)—Offset C314h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 5500082h

Bit Range	Default & Access	Field Name (ID): Description
31:16	550h RW	**TXFSTADDR_N (TXFSTADDR_N)**
15:0	82h RW	**TXFDEP_N (TXFDEP_N)**

16.1159 GTXFIFOSIZ6_0 (GTXFIFOSIZ6_0)—Offset C318h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 5D20082h

Bit Range	Default & Access	Field Name (ID): Description
31:16	5D2h RW	**TXFSTADDR_N (TXFSTADDR_N)**
15:0	82h RW	**TXFDEP_N (TXFDEP_N)**

16.1160 GTXFIFOSIZ7_0 (GTXFIFOSIZ7_0)—Offset C31Ch

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 6540082h

Bit Range	Default & Access	Field Name (ID): Description
31:16	654h RW	**TXFSTADDR_N (TXFSTADDR_N)**
15:0	82h RW	**TXFDEP_N (TXFDEP_N)**

16.1161 GTXFIFOSIZ8_0 (GTXFIFOSIZ8_0)—Offset C320h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 6D60022h

Bit Range	Default & Access	Field Name (ID): Description
31:16	6D6h RW	**TXFSTADDR_N (TXFSTADDR_N)**
15:0	22h RW	**TXFDEP_N (TXFDEP_N)**

16.1162 GTXFIFOSIZ9_0 (GTXFIFOSIZ9_0)—Offset C324h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 6F80022h

Bit Range	Default & Access	Field Name (ID): Description
31:16	6F8h RW	**TXFSTADDR_N (TXFSTADDR_N)**
15:0	22h RW	**TXFDEP_N (TXFDEP_N)**

16.1163 GTXFIFOSIZ10_0 (GTXFIFOSIZ10_0)—Offset C328h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 71A0022h

Bit Range	Default & Access	Field Name (ID): Description
31:16	71Ah RW	**TXFSTADDR_N (TXFSTADDR_N)**
15:0	22h RW	**TXFDEP_N (TXFDEP_N)**

16.1164 GTXFIFOSIZ11_0 (GTXFIFOSIZ11_0)—Offset C32Ch

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 73C0022h

Bit Range	Default & Access	Field Name (ID): Description
31:16	73Ch RW	**TXFSTADDR_N (TXFSTADDR_N)**
15:0	22h RW	**TXFDEP_N (TXFDEP_N)**

16.1165 GTXFIFOSIZ12_0 (GTXFIFOSIZ12_0)—Offset C330h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 75E0022h

Bit Range	Default & Access	Field Name (ID): Description
31:16	75Eh RW	**TXFSTADDR_N (TXFSTADDR_N)**
15:0	22h RW	**TXFDEP_N (TXFDEP_N)**

16.1166 GTXFIFOSIZ13_0 (GTXFIFOSIZ13_0)—Offset C334h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 7800022h

Bit Range	Default & Access	Field Name (ID): Description
31:16	780h RW	**TXFSTADDR_N (TXFSTADDR_N)**
15:0	22h RW	**TXFDEP_N (TXFDEP_N)**

16.1167 GTXFIFOSIZ14_0 (GTXFIFOSIZ14_0)—Offset C338h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 7A20022h

Bit Range	Default & Access	Field Name (ID): Description
31:16	7A2h RW	**TXFSTADDR_N (TXFSTADDR_N)**
15:0	22h RW	**TXFDEP_N (TXFDEP_N)**

16.1168 GTXFIFOSIZ15_0 (GTXFIFOSIZ15_0)—Offset C33Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 7C40022h

Bit Range	Default & Access	Field Name (ID): Description
31:16	7C4h RW	**TXFSTADDR_N (TXFSTADDR_N)**
15:0	22h RW	**TXFDEP_N (TXFDEP_N)**

16.1169 GRXFIFOSIZ0_0 (GRXFIFOSIZ0_0)—Offset C380h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 385h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**RXFSTADDR_N (RXFSTADDR_N)**
15:0	385h RW	**RXFDEP_N (RXFDEP_N)**

16.1170 GEVNTADRLO_0 (GEVNTADRLO_0)—Offset C400h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**EVNTADRLO (EVNTADRLO)**

16.1171 GEVNTADRHI_0 (GEVNTADRHI_0)—Offset C404h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**EVNTADRHI (EVNTADRHI)**

16.1172 GEVNTSIZ_0 (GEVNTSIZ_0)—Offset C408h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**EVNTINTRPTMASK (EVNTINTRPTMASK)**
30:16	0h RO	**reserved_31_16 (reserved_31_16)**
15:0	0h RW	**EVENTSIZ (EVENTSIZ)**

16.1173 GEVNTCOUNT_0 (GEVNTCOUNT_0)—Offset C40Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**reserved_32_16 (reserved_32_16)**
15:0	0h NA	**EVNTCOUNT (EVNTCOUNT)**

16.1174 GHWPARAMS8 (GHWPARAMS8)—Offset C600h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 2F6h

Bit Range	Default & Access	Field Name (ID): Description
31:0	2F6h RO	**DWC_USB3_DCACHE_DEPTH_INFO_32_0 (DWC_USB3_DCACHE_DEPTH_INFO_32_0)**

16.1175 GTXFIFOPRIDEV (GTXFIFOPRIDEV)—Offset C610h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**reserved_32_16 (reserved_32_16)**
15:0	0h RW	**gtxfifopridev (gtxfifopridev)**

16.1176 GFLADJ (GFLADJ)—Offset C630h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GFLADJ_REFCLK_240MHZDECR_PLS1 (GFLADJ_REFCLK_240MHZDECR_PLS1)**
30:24	0h RW	**GFLADJ_REFCLK_240MHZ_DECR (GFLADJ_REFCLK_240MHZ_DECR)**
23	0h RW	**GFLADJ_REFCLK_LPM_SEL (GFLADJ_REFCLK_LPM_SEL)**
22	0h RW	**reserved_22 (reserved_22)**
21:8	0h RW	**GFLADJ_REFCLK_FLADJ (GFLADJ_REFCLK_FLADJ)**
7	0h RW	**GFLADJ_30MHZ_SDBND_SEL (GFLADJ_30MHZ_SDBND_SEL)**
6	0h RW	**reserved_6 (reserved_6)**
5:0	0h RW	**GFLADJ_30MHZ (GFLADJ_30MHZ)**

16.1177 RID - Bridge revision ID Register (RID)—Offset 30h

register contains the Bridge Revision ID

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved0:** Reserved
7:0	0h RO	**RID:** Revision ID of the Bridge.

16.1178 GEN_REGRW1 - General Purpose register (GEN_REGRW1)—Offset 600h

General Purpose PRV RW Register 1.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW1:** This register value is brought out as oob_gen_prv_rw_reg1 out of band signal

16.1179 GEN_REGRW2 - General Purpose register (GEN_REGRW2)—Offset 604h

General Purpose PRV RW Register 2.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW2:** This register value is brought out as oob_gen_prv_rw_reg2 out of band signal

16.1180 GEN_REGRW3 - General Purpose register (GEN_REGRW3)—Offset 608h

General Purpose PRV RW Register 3.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW3:** This register value is brought out as oob_gen_prv_rw_reg3 out of band signal

16.1181 GEN_REGRW4 - General Purpose register (GEN_REGRW4)—Offset 60Ch

General Purpose PRV RW Register 4.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW4:** This register value is brought out as oob_gen_prv_rw_reg4 out of band signal

16.1182 (DEVVENDID)—Offset 0h

DEVICEVENDORID - Device ID and Vendor ID Register

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 8086h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**DEVICEID**
15:0	8086h RO	**VENDORID**

16.1183 (STATUSCOMMAND)—Offset 4h

STATUSCOMMAND- Status and Command

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 100000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved0**
29	0h RW/1C	**RMA**
28	0h RW/1C	**RTA**
27:21	0h RO	**Reserved1**
20	1h RO	**CAPLIST**
19	0h RO	**INTR_STATUS**
18:16	0h RO	**Reserved2**
15:11	0h RO	**Reserved3**
10	0h RW	**INTR_DISABLE**
9	0h RO	**Reserved4**
8	0h RW	**SERR_ENABLE**
7:3	0h RO	**Reserved5**
2	0h RW	**BME**
1	0h RW	**MSE**
0	0h RO	**Reserved6**

16.1184 (REVCLASSCODE)—Offset 8h

REVCLASSCODE - Revision ID and Class Code

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: C03FE00h

Bit Range	Default & Access	Field Name (ID): Description
31:8	C03FEh RO	**CLASS_CODES**
7:0	0h RO	**RID**

16.1185 (CLLATHEADERBIST)—Offset Ch

CLLATHEADERBIST - Cache Line Latency Header and BIST

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved0**
23	0h RO	**MULFNDEV**
22:16	0h RO	**HEADERTYPE**
15:8	0h RO	**LATTIMER**
7:0	0h RW	**CACHELINE_SIZE**

16.1186 (BAR)—Offset 10h

BAR -Base Address Register

Access Method

Type: MEM Register
(Size: 64 bits)

Device:
Function:

Default: 4h

Bit Range	Default & Access	Field Name (ID): Description
63:21	0h RW	**BASEADDR**
20:12	0h RO	**RSVD**
11:4	0h RO	**SIZEINDICATOR**

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO	**PREFETCHABLE**
2:1	2h RO	**TYPE**
0	0h RO	**MESSAGE_SPACE**

16.1187 (BAR1)—Offset 18h

BAR1 -Base Address Register1

Access Method

Type: MEM Register
(Size: 64 bits)

Device:
Function:

Default: 4h

Bit Range	Default & Access	Field Name (ID): Description
63:32	0h RW	**BASEADDR1_HIGH**
31:12	0h RW	**BASEADDR1**
11:4	0h RO	**SIZEINDICATOR1**
3	0h RO	**PREFETCHABLE1**
2:1	2h RO	**TYPE1**
0	0h RO	**MESSAGE_SPACE1**

16.1188 (SUBSYSTEMID)—Offset 2Ch

SUBSYSTEMID -Subsystem Vendor and Subsystem ID

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**SUBSYSTEMID**

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW/O	**SUBSYSTEMVENDORID**

16.1189 (EXPANSION_ROM_BASEADDR)—Offset 30h

EXPANSION ROM base address

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**EXPANSION_ROM_BASE**

16.1190 (CAPABILITYPTR)—Offset 34h

CAPABILITYPTR - Capabilities Pointer

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 80h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved0**
7:0	80h RO	**CAPPTR_POWER**

16.1191 (INTERRUPTREG)—Offset 3Ch

INTERRUPTREG - Interrupt Register

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**MAX_LAT**
23:16	0h RO	**MIN_GNT**
15:12	0h RO	**Reserved0**
11:8	1h RO	**INTPIN**
7:0	0h RW	**INTLINE**

16.1192 (POWERCAPID)—Offset 80h

POWERCAPID - PowerManagement Capability ID

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 48039001h

Bit Range	Default & Access	Field Name (ID): Description
31:27	9h RO	**PMESUPPORT**
26:19	0h RO	**Reserved0**
18:16	3h RO	**VERSION**
15:8	90h RO	**NXTCAP**
7:0	1h RO	**POWER_CAP**

16.1193 (PMECTRLSTATUS)—Offset 84h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 8h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved0**

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**PMESTATUS**
14:9	0h RO	**Reserved1**
8	0h RW	**PMEENABLE**
7:4	0h RO	**Reserved2**
3	1h RO	**NO_SOFT_RESET**
2	0h RO	**Reserved3**
1:0	0h RW	**POWERSTATE**

16.1194 (PCIDEVIDLE_CAP_RECORD)—Offset 90h

PCI DEVICE IDLE CAPABILITY RECORD

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: F0140009h

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RO	**VEND_CAP**
27:24	0h RO	**REVID**
23:16	14h RO	**CAP_LENGTH**
15:8	0h RO	**NEXT_CAP**
7:0	9h RO	**CAPID**

16.1195 (DEVID_VEND_SPECIFIC_REG)—Offset 94h

DEVID VENDOR SPECIFIC REG

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 1400010h

Bit Range	Default & Access	Field Name (ID): Description
31:20	14h RO	**VSEC_LENGTH**
19:16	0h RO	**VSEC_REV**
15:0	10h RO	**VSECID**

16.1196 (D0I3_CONTROL_SW_LTR_MMIO_REG)—Offset 98h

SW LTR Update MMIO Location Register

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**SW_LAT_DWORD_OFFSET**
3:1	0h RO	**SW_LAT_BAR_NUM**
0	0h RO	**SW_LAT_VALID**

16.1197 (DEVICE_IDLE_POINTER_REG)—Offset 9Ch

Device IDLE pointer register

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 10F8301h

Bit Range	Default & Access	Field Name (ID): Description
31:4	10F830h RO	**DWORD_OFFSET**
3:1	0h RO	**BAR_NUM**
0	1h RO	**VALID**

16.1198 (D0I3_MAX_POW_LAT_PG_CONFIG)—Offset A0h

DEVICE PG CONFIG

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 80800h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved0**
21	0h RW	**HAE**
20	0h RO	**Reserved1**
19	1h RW	**SLEEP_EN**
18	0h RW	**PGE**
17	0h RW	**I3_ENABLE**
16	0h RW	**D3_ENABLE**
15:13	0h RO	**Reserved2**
12:10	2h RW/O	**POW_LAT_SCALE**
9:0	0h RW/O	**POW_LAT_VALUE**

16.1199 (MANID)—Offset F8h

Manufacturers ID

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**MANID**

16.1200 (DEVVENDID)—Offset 0h

DEVICEVENDORID - Device ID and Vendor ID Register

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 21
Function: 1

Default: 8086h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**DEVICEID**
15:0	8086h RO	**VENDORID**

16.1201 (STATUSCOMMAND)—Offset 4h

STATUSCOMMAND- Status and Command

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 21
Function: 1

Default: 100000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved0**
29	0h RW/1C	**RMA**
28	0h RW/1C	**RTA**
27:21	0h RO	**Reserved1**
20	1h RO	**CAPLIST**
19	0h RO	**INTR_STATUS**
18:16	0h RO	**Reserved2**
15:11	0h RO	**Reserved3**
10	0h RW	**INTR_DISABLE**
9	0h RO	**Reserved4**

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW	**SERR_ENABLE**
7:3	0h RO	**Reserved5**
2	0h RW	**BME**
1	0h RW	**MSE**
0	0h RO	**Reserved6**

16.1202 (REVCLASSCODE)—Offset 8h

REVCLASSCODE - Revision ID and Class Code

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 1

Default: C03FE00h

Bit Range	Default & Access	Field Name (ID): Description
31:8	C03FEh RO	**CLASS_CODES**
7:0	0h RO	**RID**

16.1203 (CLLATHEADERBIST)—Offset Ch

CLLATHEADERBIST - Cache Line Latency Header and BIST

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 1

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved0**
23	0h RO	**MULFNDEV**
22:16	0h RO	**HEADERTYPE**

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	**LATTIMER**
7:0	0h RW	**CACHELINE_SIZE**

16.1204 (BAR)—Offset 10h

BAR -Base Address Register

Access Method

Type: CFG Register
(Size: 64 bits)

Device: 21
Function: 1

Default: 4h

Bit Range	Default & Access	Field Name (ID): Description
63:21	0h RW	**BASEADDR**
20:12	0h RO	**RSVD**
11:4	0h RO	**SIZEINDICATOR**
3	0h RO	**PREFETCHABLE**
2:1	2h RO	**TYPE**
0	0h RO	**MESSAGE_SPACE**

16.1205 (BAR1)—Offset 18h

BAR1 -Base Address Register1

Access Method

Type: CFG Register
(Size: 64 bits)

Device: 21
Function: 1

Default: 4h

Bit Range	Default & Access	Field Name (ID): Description
63:32	0h RW	**BASEADDR1_HIGH**
31:12	0h RW	**BASEADDR1**

Bit Range	Default & Access	Field Name (ID): Description
11:4	0h RO	**SIZEINDICATOR1**
3	0h RO	**PREFETCHABLE1**
2:1	2h RO	**TYPE1**
0	0h RO	**MESSAGE_SPACE1**

16.1206 (SUBSYSTEMID)—Offset 2Ch

SUBSYSTEMID -Subsystem Vendor and Subsystem ID

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 21
Function: 1

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**SUBSYSTEMID**
15:0	0h RW/O	**SUBSYSTEMVENDORID**

16.1207 (EXPANSION_ROM_BASEADDR)—Offset 30h

EXPANSION ROM base address

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 21
Function: 1

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**EXPANSION_ROM_BASE**

16.1208 (CAPABILITYPTR)—Offset 34h

CAPABILITYPTR - Capabilities Pointer

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 21
Function: 1

Default: 80h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved0**
7:0	80h RO	**CAPPTR_POWER**

16.1209 (INTERRUPTREG)—Offset 3Ch

INTERRUPTREG - Interrupt Register

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 21
Function: 1

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**MAX_LAT**
23:16	0h RO	**MIN_GNT**
15:12	0h RO	**Reserved0**
11:8	1h RO	**INTPIN**
7:0	0h RW	**INTLINE**

16.1210 (POWERCAPID)—Offset 80h

POWERCAPID - PowerManagement Capability ID

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 21
Function: 1

Default: 48039001h

Bit Range	Default & Access	Field Name (ID): Description
31:27	9h RO	**PMESUPPORT**

Bit Range	Default & Access	Field Name (ID): Description
26:19	0h RO	**Reserved0**
18:16	3h RO	**VERSION**
15:8	90h RO	**NXTCAP**
7:0	1h RO	**POWER_CAP**

16.1211 (PMECTRLSTATUS)—Offset 84h

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 21
Function: 1

Default: 8h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved0**
15	0h RW/1C	**PMESTATUS**
14:9	0h RO	**Reserved1**
8	0h RW	**PMEENABLE**
7:4	0h RO	**Reserved2**
3	1h RO	**NO_SOFT_RESET**
2	0h RO	**Reserved3**
1:0	0h RW	**POWERSTATE**

16.1212 (PCIDEVIDLE_CAP_RECORD)—Offset 90h

PCI DEVICE IDLE CAPABILITY RECORD

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 21
Function: 1

Default: F0140009h

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RO	**VEND_CAP**
27:24	0h RO	**REVID**
23:16	14h RO	**CAP_LENGTH**
15:8	0h RO	**NEXT_CAP**
7:0	9h RO	**CAPID**

16.1213 (DEVID_VEND_SPECIFIC_REG)—Offset 94h

DEVID VENDOR SPECIFIC REG

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 1

Default: 1400010h

Bit Range	Default & Access	Field Name (ID): Description
31:20	14h RO	**VSEC_LENGTH**
19:16	0h RO	**VSEC_REV**
15:0	10h RO	**VSECID**

16.1214 (D0I3_CONTROL_SW_LTR_MMIO_REG)—Offset 98h

SW LTR Update MMIO Location Register

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 1

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**SW_LAT_DWORD_OFFSET**
3:1	0h RO	**SW_LAT_BAR_NUM**

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**SW_LAT_VALID**

16.1215 (DEVICE_IDLE_POINTER_REG)—Offset 9Ch

Device IDLE pointer register

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 1

Default: 10F8301h

Bit Range	Default & Access	Field Name (ID): Description
31:4	10F830h RO	**DWORD_OFFSET**
3:1	0h RO	**BAR_NUM**
0	1h RO	**VALID**

16.1216 (D0I3_MAX_POW_LAT_PG_CONFIG)—Offset A0h

DEVICE PG CONFIG

Access Method

Type: CFG Register (Size: 32 bits)
Device: 21
Function: 1

Default: 80800h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved0**
21	0h RW	**HAE**
20	0h RO	**Reserved1**
19	1h RW	**SLEEP_EN**
18	0h RW	**PGE**
17	0h RW	**I3_ENABLE**

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW	**D3_ENABLE**
15:13	0h RO	**Reserved2**
12:10	2h RW/O	**POW_LAT_SCALE**
9:0	0h RW/O	**POW_LAT_VALUE**

16.1217 (MANID)—Offset F8h

Manufacturers ID

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 21
Function: 1

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**MANID**

§ §

17 USB-Device (xDCI) Registers

This chapter documents the registers in Bus: 0, Device 21, Function 1.

17.1 DEVVENDID — Offset 0h

DEVICEVENDORID - Device ID and Vendor ID Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + 0h	8086 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**DEVICEID:**
15:0	8086h RO	**VENDORID:**

17.2 STATUSCOMMAND — Offset 4h

STATUSCOMMAND- Status and Command

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + 4h	100000 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**RESERVED0:** Reserved
29	0h RW/1C	**RMA:**
28	0h RW/1C	**RTA:**
27:21	0h RO	**RESERVED1:** Reserved
20	1h RO	**CAPLIST:**
19	0h RO	**INTR_STATUS:**
18:16	0h RO	**RESERVED2:** Reserved
15:11	0h RO	**RESERVED3:** Reserved

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW	**INTR_DISABLE:**
9	0h RO	**RESERVED4:** Reserved
8	0h RW	**SERR_ENABLE:**
7:3	0h RO	**RESERVED5:** Reserved
2	0h RW	**BME:**
1	0h RW	**MSE:**
0	0h RO	**RESERVED6:** Reserved

17.3 REVCLASSCODE — Offset 8h

REVCLASSCODE - Revision ID and Class Code

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + 8h	C03FE00 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	C03FEh RO	**CLASS_CODES:**
7:0	0h RO	**RID:**

17.4 CLLATHEADERBIST — Offset Ch

CLLATHEADERBIST - Cache Line Latency Header and BIST

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**RESERVED0:** Reserved
23	0h RO	**MULFNDEV:**

Bit Range	Default & Access	Field Name (ID): Description
22:16	0h RO	**HEADERTYPE:**
15:8	0h RO	**LATTIMER:**
7:0	0h RW	**CACHELINE_SIZE:**

17.5 BAR — Offset 10h

BAR -Base Address Register

Type	Size	Offset	Default
CFG	64 bit	[B:0, D:21, F:1] + 10h	4 h

Bit Range	Default & Access	Field Name (ID): Description
63:21	0h RW	**BASEADDR:**
20:12	0h RO	**RSVD:** Reserved
11:4	0h RO	**SIZEINDICATOR:**
3	0h RO	**PREFETCHABLE:**
2:1	2h RO	**TYPE:**
0	0h RO	**MESSAGE_SPACE:**

17.6 BAR1 — Offset 18h

BAR1 -Base Address Register1

Type	Size	Offset	Default
CFG	64 bit	[B:0, D:21, F:1] + 18h	4 h

Bit Range	Default & Access	Field Name (ID): Description
63:32	0h RW	**BASEADDR1_HIGH:**
31:12	0h RW	**BASEADDR1:**

Bit Range	Default & Access	Field Name (ID): Description
11:4	0h RO	**SIZEINDICATOR1:**
3	0h RO	**PREFETCHABLE1:**
2:1	2h RO	**TYPE1:**
0	0h RO	**MESSAGE_SPACE1:**

17.7 SUBSYSTEMID — Offset 2Ch

SUBSYSTEMID -Subsystem Vendor and Subsystem ID

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**SUBSYSTEMID:**
15:0	0h RW/O	**SUBSYSTEMVENDORID:**

17.8 EXPANSION_ROM_BASEADDR — Offset 30h

EXPANSION ROM base address

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + 30h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**EXPANSION_ROM_BASE:**

17.9 CAPABILITYPTR — Offset 34h

CAPABILITYPTR - Capabilities Pointer

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + 34h	80 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**RESERVED0:** Reserved
7:0	80h RO	**CAPPTR_POWER:**

17.10 INTERRUPTREG — Offset 3Ch

INTERRUPTREG - Interrupt Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + 3Ch	100 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**MAX_LAT:**
23:16	0h RO	**MIN_GNT:**
15:12	0h RO	**RESERVED0:** Reserved
11:8	1h RO	**INTPIN:**
7:0	0h RW	**INTLINE:**

17.11 POWERCAPID — Offset 80h

POWERCAPID - PowerManagement Capability ID

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + 80h	48039001 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	9h RO	**PMESUPPORT:**
26:19	0h RO	**RESERVED0:** Reserved
18:16	3h RO	**VERSION:**
15:8	90h RO	**NXTCAP:**

Bit Range	Default & Access	Field Name (ID): Description
7:0	1h RO	**POWER_CAP:**

17.12 PMECTRLSTATUS — Offset 84h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + 84h	8 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED0:** Reserved
15	0h RW/1C	**PMESTATUS:**
14:9	0h RO	**RESERVED1:** Reserved
8	0h RW	**PMEENABLE:**
7:4	0h RO	**RESERVED2:** Reserved
3	1h RO	**NO_SOFT_RESET:**
2	0h RO	**RESERVED3:** Reserved
1:0	0h RW	**POWERSTATE:**

17.13 PCIDEVIDLE_CAP_RECORD — Offset 90h

PCI DEVICE IDLE CAPABILITY RECORD

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + 90h	F0140009 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RO	**VEND_CAP:**
27:24	0h RO	**REVID:**
23:16	14h RO	**CAP_LENGTH:**

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	**NEXT_CAP:**
7:0	9h RO	**CAPID:**

17.14 DEVID_VEND_SPECIFIC_REG — Offset 94h

DEVID VENDOR SPECIFIC REG

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + 94h	1400010 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	14h RO	**VSEC_LENGTH:**
19:16	0h RO	**VSEC_REV:**
15:0	10h RO	**VSECID:**

17.15 D0I3_CONTROL_SW_LTR_MMIO_REG — Offset 98h

SW LTR Update MMIO Location Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + 98h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**SW_LAT_DWORD_OFFSET:**
3:1	0h RO	**SW_LAT_BAR_NUM:**
0	0h RO	**SW_LAT_VALID:**

17.16 DEVICE_IDLE_POINTER_REG — Offset 9Ch

Device IDLE pointer register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + 9Ch	10F8301 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	10F830h RO	**DWORD_OFFSET:**
3:1	0h RO	**BAR_NUM:**
0	1h RO	**VALID:**

17.17 D0I3_MAX_POW_LAT_PG_CONFIG — Offset A0h

DEVICE PG CONFIG

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + A0h	80800 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**RESERVED0:** Reserved
21	0h RW	**HAE:**
20	0h RO	**RESERVED1:** Reserved
19	1h RW	**SLEEP_EN:**
18	0h RW	**PGE:**
17	0h RW	**I3_ENABLE:**
16	0h RW	**D3_ENABLE:**
15:13	0h RO	**RESERVED2:** Reserved
12:10	2h RW/O	**POW_LAT_SCALE:**
9:0	0h RW/O	**POW_LAT_VALUE:**

17.18 GEN_REGRW1 — Offset B0h

General Purpose Read Write Register1

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + B0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW1:**

17.19 GEN_REGRW2 — Offset B4h

General Purpose Read Write Register2

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + B4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW2:**

17.20 GEN_REGRW3 — Offset B8h

General Purpose Read Write Register3

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + B8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW3:**

17.21 GEN_REGRW4 — Offset BCh

General Purpose Read Write Register4

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + BCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RW	**GEN_REG_RW4:** Reserved
2	0h RW	**XDCI_PIPE_CLK_GATE_DIS:** xdci pipe clock gating disable
1	0h RW	**XDCI_SB_CLK_GATE_DIS:** xdci sideband clock gating disable
0	0h RW	**XDCI_PRIM_CLK_GATE_DIS:** xdci primary clock gating disable

17.22 GEN_INPUT_REGRW — Offset C0h

General Purpose Input Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:21, F:1] + C0h	1008000B h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**BVALID_HW (HW_BVALID):** hardware bvalid
30	0h RO	**GSTS_BUSERRADDVLD:** Bus error
29	0h RO	**DRD_MODE:** SS DRD mode 0 device mode 1 host mode
28	1h RO	**IDPIN:** IDPIN value (debounce)
27:26	0h RO	**U2_DSSR_STATE:** USB2 DSSR State
25:21	0h RO	**U2_PRT_STATE:** USB2 Port State
20:17	4h RO	**LTDB_LINK_STATE:** USB3 Link state
16:13	0h RO	**LTDB_SUB_STATE:** USB3 Link sub state
12	0h RO	**RESERVED1:** Reserved
11:10	0h RO	**CURRENT_POWER_STATE_U3PMU:** The current power state of the core. When equal to '3', the PMU is controlling the PHY, and when equal to '0', the core is controlling the PHY
9:8	0h RO	**CURRENT_POWER_STATE_U2PMU:** The current power state of the core. When equal to '3', the PMU is controlling the PHY, and when equal to '0', the core is controlling the PHY
7	0h RO	**CONNECT_STATE_U3PMU:** When '1', indicates the PMU is maintaining at least one connection to the host or a device (the link is in U3). When '0', indicates the PMU has no connection to the host or any device.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RO	**CONNECT_STATE_U2PMU:** When '1', indicates the PMU is maintaining at least one connection to the host or a device (the link is in L1, L2). When '0', indicates the PMU has no connection to the host or any device
5	0h RO	**UTMI_L1_SUSPEND_COM_N:** Common L1 suspend
4	0h RO	**UTMI_SUSPEND_COM_N:** Common suspend
3	1h RO	**UTMI_SUSPEND_N:** USB 2.0 Port Suspend
2:1	1h RO	**USB2_ENUMSPEED:** Device Enumerated Speed
0	1h RO	**B2RL_CUR_MODE:** Current Mode. Current Mode 1'b0 Host ,1'b1 Device

17.23 MANID — Offset F8h

Manufacturers ID register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + F8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**MANID:** Manufacturer ID: Default value comes from straps.

§ §

18 PCI Express* Controller Registers

This chapter documents the registers of the PCIe controller devices. The processor contains multiple PCIe controller devices:

- Bus: 0, Device: 20, Function: 0 PCIe0 (Func0)
- Bus: 0, Device: 20, Function: 1 PCIe0 (Func1)
- Bus: 0, Device: 19, Function: 0 PCIe1 (Func0)
- Bus: 0, Device: 19, Function: 1 PCIe1 (Func1)
- Bus: 0, Device: 19, Function: 2 PCIe1 (Func2)
- Bus: 0, Device: 19, Function: 3 PCIe1 (Func3)

NOTE: Register default values are taken from device PCIe0 only. Refer Vol1 for Device IDs

18.1 ID — Offset 0h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO/V	**Device Identification (DID):** The value of this ID is product specific.
15:0	0h RO	**Vendor Identification (VID):** Indicates Intel

18.2 Device Command; Primary Status (CMD_PSTS) — Offset 4h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C/V	**DPE:** Set when the root port receives a command or data from the backbone with a parity error. This is set even if CMD.PERE is not set.
30	0h RW/1C/V	**Signaled System Error (SSE):** Set when the root port signals a system error to the internal SERR# logic.
29	0h RW/1C/V	**Received Master Abort (RMA):** Set when the root port receives a completion with unsupported request status from the backbone.

Bit Range	Default & Access	Field Name (ID): Description
28	0h RW/1C/V	**Received Target Abort (RTA):** Set when the root port receives a completion with completer abort from the backbone.
27	0h RW/1C/V	**Signaled Target Abort (STA):** Set whenever the root port forwards a target abort received from the downstream device onto the backbone.
26:25	0h RO	**Primary DEVSEL# Timing Status (PDTS):** Reserved per PCI-Express spec
24	0h RW/1C/V	**Master Data Parity Error Detected (DPD):** Set when the root port receives a completion with a data parity error on the backbone and CMD.PERE is set.
23	0h RO	**Primary Fast Back to Back Capable (PFBC):** Reserved per PCI-Express spec.
22	0h RO	**Reserved (RSVD):** Reserved
21	0h RO	**Primary 66 MHz Capable (PC66):** Reserved per PCI-Express spec.
20	0h RO	**Capabilities List (CLIST):** Indicates the presence of a capabilities list.
19	0h RO/V	**Interrupt Status (IS):** Indicates status of hot plug and power management interrupts on the root port that result in INTx# message generation. This bit is not set if MSI is enabled. If MSI is not enabled, this bit is set regardless of the state of CMD.ID.
18:16	0h RO	**Reserved (RSVD_1):** Reserved
15:11	0h RO	**Reserved (RSVD_2):** Reserved
10	0h RW/V2	**Interrupt Disable (ID):** This disables pin-based INTx# interrupts on enabled hot plug and power management events. This bit has no effect on MSI operation. When set, internal INTx# messages will not be generated. When cleared, internal INTx# messages are generated if there is an interrupt for hot plug or power management and MSI is not enabled. This bit does not affect interrupt forwarding from devices connected to the root port. Assert_INTx and Deassert_INTx messages will still be forwarded to the internal interrupt controllers if this bit is set. For PCI Bus Emulation Mode compatibility, if the PCIBEM register is set, this register is RO and returns a value of 0 when read, else it is RW with the functionality described above.
9	0h RO	**Fast Back to Back Enable (FBE):** Reserved per PCI-Express spec.
8	0h RW	**SERR# Enable (SEE):** When set, enables the root port to generate an SERR# message when PSTS.SSE is set.
7	0h RO	**Wait Cycle Control (WCC):** Reserved per PCI-Express spec.
6	0h RW	**Parity Error Response Enable (PERE):** Indicates that the device is capable of reporting parity errors as a master on the backbone.
5	0h RO	**VGA Palette Snoop (VGA_PSE):** Reserved per PCI-Express spec.
4	0h RO	**Memory Write and Invalidate Enable (MWIE):** Reserved per PCI-Express spec.
3	0h RO	**Special Cycle Enable (SCE):** Reserved per PCI-Express and PCI bridge spec.
2	0h RW	**Bus Master Enable (BME):** When set, allows the root port to forward Memory and I/O Read/Write cycles onto the backbone from a PCI-Express device. When this bit is 0b, Memory and I/O requests received at a Root Port must be handled as Unsupported Requests (UR). This bit does not affect forwarding of Completions in either the Upstream or Downstream direction. The forwarding of Requests other than Memory or I/O requests is not controlled by this bit.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Memory Space Enable (MSE):** When set, memory cycles within the range specified by the memory base and limit registers can be forwarded to the PCI-Express device. When cleared, these memory cycles are master aborted on the backbone.
0	0h RW	**I/O Space Enable (IOSE):** When set, I/O cycles within the range specified by the I/O base and limit registers can be forwarded to the PCI-Express device. When cleared, these cycles are master aborted on the backbone..

18.3 Revision ID;Class Code (RID_CC) — Offset 8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Base Class Code (BCC):** Indicates the device is a bridge device.
23:16	0h RO/V	**Sub-Class Code (SCC):** The default indicates the device is a PCI-to-PCI bridge. If the MPC.BT register is set to a '1' for a Host Bridge, this register reads 00h.
15:8	0h RO/V	**Programming Interface (PI):** The value reported in this register is a function of the Decode Control.Subtractive Decode Enable (SDE) register. SDE Value reported in this register 0: 00h 1: 01h
7:0	0h RO/V	**Revision ID (RID):** Indicates the revision of the bridge.

18.4 Cache Line Size; Primary Latency Timer; Header Type (CLS_PLT_HTYPE) — Offset Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23	0h RO	**Multi-function Device (MFD):** This bit is '1' to indicate a multi-function device.
22:16	0h RO/V	**Header Type (HTYPE):** The default mode identifies the header layout of the configuration space, which is a PCI-to-PCI bridge. If the MPC.BT register is set to a '1' for a Host Bridge, this register reads 00h.
15:11	0h RO	**Latency Count (CT):** Reserved per PCI-Express spec
10:8	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Line Size (LS):** This is read/write but contains no functionality, per PCI-Express spec

18.5 Bus Numbers; Secondary Latency Timer (BNUM_SLT) — Offset 18h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 18h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW/V2	**Secondary Latency Timer (SLT):** For PCI Bus Emulation Mode compatibility, if the DC.PCIBEM register is set, this register is a RW register; else this register is RO and returns 0. This register does not affect the behavior of any HW logic.
23:16	0h RW	**Subordinate Bus Number (SBBN):** Indicates the highest PCI bus number below the bridge.
15:8	0h RW	**Secondary Bus Number (SCBN):** Indicates the bus number the port.
7:0	0h RW	**Primary Bus Number (PBN):** Indicates the bus number of the backbone.

18.6 I/O Base and Limit; Secondary Status (IOBL_SSTS) — Offset 1Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 1Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C/V	**Detected Parity Error (DPE):** Set when the port receives a poisoned TLP.
30	0h RW/1C/V	**Received System Error (RSE):** Set when the port receives an ERR_FATAL or ERR_NONFATAL message from the device.
29	0h RW/1C/V	**Received Master Abort (RMA):** Set when the port receives a completion with 'Unsupported Request' status from the device.
28	0h RW/1C/V	**Received Target Abort (RTA):** Set when the port receives a completion with 'Completion Abort' status from the device.
27	0h RW/1C/V	**Signaled Target Abort (STA):** Set when the port generates a completion with 'Completion Abort' status to the device.
26:25	0h RO/V	**Secondary DEVSEL# Timing Status (SDTS):** Reserved per PCI-Express spec For PCI Bus Emulation Mode compatibility, if the PCIBEM register is set, this register returns a value of 01b when read, else this register returns a value of 00b.

Bit Range	Default & Access	Field Name (ID): Description
24	0h RW/1C/V	**Data Parity Error Detected (DPD):** Set when the BCTRL.PERE, and either of the following two conditions occurs: Port receives completion marked poisoned. Port poisons a write request to the secondary side.
23	0h RO/V	**Secondary Fast Back to Back Capable (SFBC):** Reserved per PCI Express spec For PCI Bus Emulation Mode compatibility, if the PCIBEM register is set, this register returns a value of 1b when read, else this register returns a value of 0b.
22	0h RO	**Reserved (RSVD):** Reserved
21	0h RO	**Secondary 66 MHz Capable (SC66):** Reserved per PCI Express spec
20:16	0h RO	**Reserved (RSVD_1):** Reserved
15:12	0h RW	**I/O Address Limit (IOLA):** I/O Base bits corresponding to address lines 15:12 for 4KB alignment. Bits 11:0 are assumed to be padded to FFFh.
11:8	0h RO	**I/O Limit Address Capability (IOLC):** Indicates that the bridge does not support 32-bit I/O addressing.
7:4	0h RW	**I/O Base Address (IOBA):** I/O Base bits corresponding to address lines 15:12 for 4KB alignment. Bits 11:0 are assumed to be padded to 000h.
3:0	0h RO	**I/O Base Address Capability (IOBC):** Indicates that the bridge does not support 32-bit I/O addressing.

18.7 Memory Base and Limit (MBL) — Offset 20h

Accesses that are within the ranges specified in this register will be sent to the attached device if CMD.MSE is set. Accesses from the attached device that are outside the ranges specified will be forwarded to the backbone if CMD.BME is set. The comparison performed is MB [gt]= AD[lb]31:20[rb] [lt]= ML.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 20h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW	**Memory Limit (ML):** These bits are compared with bits 31:20 of the incoming address to determine the upper 1MB aligned value of the range.
19:16	0h RO	**Reserved (RSVD):** Reserved
15:4	0h RW	**Memory Base (MB):** These bits are compared with bits 31:20 of the incoming address to determine the lower 1MB aligned value of the range.
3:0	0h RO	**Reserved (RSVD_1):** Reserved

18.8 Prefetchable Memory Base and Limit (PMBL) — Offset 24h

Accesses that are within the ranges specified in this register will be sent to the device if CMD.MSE is set. Accesses from the device that are outside the ranges specified will be forwarded to the backbone if CMD.BME is set. The comparison performed is PMBU32:PMB [gt]= AD[lb]63:32[rb]:AD[lb]31:20[rb] [lt]= PMLU32:PML.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 24h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW	**Prefetchable Memory Limit (PML):** These bits are compared with bits 31:20 of the incoming address to determine the upper 1MB aligned value of the range.
19:16	0h RO	**64-bit Indicator (I64L):** Indicates support for 64-bit addressing.
15:4	0h RW	**Prefetchable Memory Base (PMB):** These bits are compared with bits 31:20 of the incoming address to determine the lower 1MB aligned value of the range.
3:0	0h RO	**64-bit Indicator (I64B):** Indicates support for 64-bit addressing.

18.9 Prefetchable Memory Base Upper 32 Bits (PMBU32) — Offset 28h

Size:32 bits

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 28h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Prefetchable Memory Base Upper Portion (PMBU):** Upper 32-bits of the prefetchable address base.

18.10 Prefetchable Memory Limit Upper 32 Bits (PMLU32) — Offset 2Ch

Size:32 bits

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Prefetchable Memory Limit Upper Portion (PMLU):** Upper 32-bits of the prefetchable address limit.

18.11 Capabilities List Pointer (CAPP) — Offset 34h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 34h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD):** Reserved
7:0	0h RW/O	**Capabilities Pointer (PTR):** Indicates that the pointer for the first entry in the capabilities list. BIOS can determine which capabilities will be exposed by including or removing them from the capability linked list. As this register is RWO, BIOS must write a value to this register, even if it is to re-write the default value. Capability Linked List (Default Settings) OffsetCapability Next Pointer 40h PCI Express 80h 80h Message Signaled Interrupt (MSI) 90h 90h Subsystem Vendor A0h A0h PCI Power Management 00h Extended PCIe Capability Linked List OffsetCapability Next Pointer 100h Advanced Error Reporting 000h

18.12 Interrupt Information; Bridge Control (INTR_BCTRL) — Offset 3Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 3Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD):** Reserved
27	0h RW/V2	**Discard Timer SERR# Enable (DTSE):** Reserved per PCI-Express spec. For PCI Bus Emulation Mode compatibility, if the DC.PCIBEM register is set, this register is RW else it is RO. This register is only maintained for SW compatibility and has no functionality within the port.
26	0h RO	**Discard Timer Status (DTS):** Reserved per PCI-Express spec. For PCI Bus Emulation Mode compatibility, this register can remain RO as no secondary discard timer exists that will ever cause it to be set.
25	0h RW/V2	**Secondary Discard Timer (SDT):** Reserved per Express spec. For PCI Bus Emulation Mode compatibility, if the DC.PCIBEM register is set, this register is RW else it is RO. This register is only maintained for SW compatibility and has no functionality within the port.
24	0h RW/V2	**Primary Discard Timer (PDT):** Reserved per Express spec. For PCI Bus Emulation Mode compatibility, if the DC.PCIBEM register is set, this register is RW else it is RO. This register is only maintained for SW compatibility and has no functionality within the port.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RO	**Fast Back to Back Enable (FBE):** Reserved per Express spec.
22	0h RW	**Secondary Bus Reset (SBR):** Triggers a Hot Reset on the PCI-Express port.
21	0h RW/V2	**Master Abort Mode (MAM):** Reserved per Express spec. For PCI Bus Emulation Mode compatibility, if the DC.PCIBEM register is set, this register is RW else it is RO. This register is only maintained for SW compatibility and has no functionality within the port.
20	0h RW	**VGA 16-Bit Decode (V16):** When set, indicates that the I/O aliases of the VGA range (see BCTRL:VE definition below), are not enabled. 0: Execute 10-bit address decode on VGA I/O accesses. 1: Execute 16-bit address decode on VGA I/O accesses.
19	0h RW	**VGA Enable (VE):** When set, the following ranges will be claimed off the backbone by the root port: Memory ranges A0000h-BFFFFh I/O ranges 3B0h 3BBh and 3C0h 3DFh, and all aliases of bits 15:10 in any combination of 1's
18	0h RW	**ISA Enable (IE):** This bit only applies to I/O addresses that are enabled by the I/O Base and I/O Limit registers and are in the first 64KB of PCI I/O space. If this bit is set, the root port will block any forwarding from the backbone to the device of I/O transactions addressing the last 768 bytes in each 1KB block (offsets 100h to 3FFh).
17	0h RW	**SE:** When set, ERR_COR, ERR_NONFATAL, and ERR_FATAL messages received are forwarded to the backbone. When cleared, they are not.
16	0h RW	**Parity Error Response Enable (PERE):** When set, poisoned write TLPs and completions indicating poisoned TLPs will set the SSTS.DPD.
15:8	0h RO/V	**Interrupt Pin (IPIN):** Indicates the interrupt pin driven by the root port. At reset, this register takes on the following values, which reflect the reset state of the STRPFUSECFG register in chipset config space: Port Bits[lb]15:12[rb] Bits[lb]11:08[rb] 1 0h STRPFUSECFG.P1IP 2 0h STRPFUSECFG.P2IP 3 0h STRPFUSECFG.P3IP 4 0h STRPFUSECFG.P4IP 5 0h STRPFUSECFG.P5IP 6 0h STRPFUSECFG.P6IP 7 0h STRPFUSECFG.P7IP 8 0h STRPFUSECFG.P8IP The value that is programmed into STRPFUSECFG.PxIP is always reflected in this register. For PCI Bus Emulation Mode compatibility, if the DC.PCIBEM register is set, this register returns a value of 00h when read.
7:0	0h RW	**Interrupt Line (ILINE):** Software written value to indicate which interrupt line (vector) the interrupt is connected to. No hardware action is taken on this register.

18.13 Capabilities List; PCI Express Capabilities (CLIST_XCAP) — Offset 40h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 40h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved
30	0h RO	**Reserved (RSVD_1):** Reserved
29:25	0h RO	**Interrupt Message Number (IMN):** The root port does not have multiple MSI interrupt numbers.
24	0h RW/O	**Slot Implemented (SI):** Indicates whether the root port is connected to a slot. Slot support is platform specific. BIOS programs this field, and it is maintained until a platform reset.

Bit Range	Default & Access	Field Name (ID): Description
23:20	0h RO	**Device / Port Type (DT):** Indicates this is a PCI-Express root port
19:16	0h RO	**Capability Version (CV):** Version 2.0 indicates devices compliant to the PCI Express 2.0 specification which incorporates the Register Expansion ECN.
15:8	0h RW/O	**NEXT:** Indicates the location of the next capability. The default value of this register is 80h which points to the MSI Capability structure. BIOS can determine which capabilities will be exposed by including or removing them from the capability linked list. As this register is RWO, BIOS must write a value to this register, even if it is to re-write the default value.
7:0	0h RO	**Capability ID (CID):** Indicates this is a PCI Express capability

18.14 Device Capabilities (DCAP) — Offset 44h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 44h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**Reserved (RSVD):** Reserved
28	0h RO	**Function Level Reset Capable (FLRC):** Not supported in Root Ports
27:26	0h RO	**Captured Slot Power Limit Scale (CSPS):** Not supported
25:18	0h RO	**Captured Slot Power Limit Value (CSPV):** Not supported
17:16	0h RO	**Reserved (RSVD_1):** Reserved
15	0h RO	**Role Based Error Reporting (RBER):** When Set, this bit indicates that the Function implements the functionality originally defined in the Error Reporting ECN for PCI Express Base Specification, Revision 1.0a, and later incorporated into PCI Express Base Specification, Revision 1.1. This bit must be Set by all Functions conforming to the ECN, PCI Express Base Specification, Revision 1.1, or subsequent PCI Express Base Specification revisions.
14	0h RO	**Reserved (RSVD_2):** Reserved
13	0h RO	**Reserved (RSVD_3):** Reserved
12	0h RO	**Reserved (RSVD_4):** Reserved
11:9	0h RO	**Endpoint L1 Acceptable Latency (E1AL):** Reserved for root ports.
8:6	0h RO	**Endpoint L0 Acceptable Latency (E0AL):** Reserved for Root port.
5	0h RO	**Extended Tag Field Supported (ETFS):** The root port never needs to initiate a transaction as a Requester with the Extended Tag bits being set. This bit does not affect the root port's ability to forward requests as a bridge as the root port always supports forwarding requests with extended tags.

Bit Range	Default & Access	Field Name (ID): Description
4:3	0h RO	**Phantom Functions Supported (PFS):** No phantom functions supported
2:0	0h RW/O	**Max Payload Size Supported (MPS):** BIOS should write to this field during system initialization. Only Max Payload Size of up to 256B is supported. Programming this field to any values other than 128B max payload size will result in aliasing to 128B max payload size. 000b: 128 bytes max payload size. 001b: 256 bytes max payload size. 010b: 512 bytes max payload size. 011b: 1024 bytes max payload size. 100b: 2048 bytes max payload size. 101b: 4096 bytes max payload size. 110b: Reserved. 111b: Reserved. This field applies only to the PCIe link interface.

18.15 Device Control; Device Status (DCTL_DSTS) — Offset 48h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 48h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (RSVD):** Reserved
21	0h RO	**Transactions Pending (TDP):** This bit has no meaning for the root port since it never initiates a non-posted request with its own Requester ID.
20	0h RO	**AUX Power Detected (APD):** The root port contains AUX power for wakeup
19	0h RW/1C/V	**Unsupported Request Detected (URD):** Indicates an unsupported request was detected.
18	0h RW/1C/V	**Fatal Error Detected (FED):** Indicates a fatal error was detected. Set when a fatal error occurred on from a data link protocol error, buffer overflow, or malformed tlp
17	0h RW/1C/V	**Non-Fatal Error Detected (NFED):** Indicates a non-fatal error was detected. Set when an received a non-fatal error occurred from a poisoned tlp, unexpected completions, unsupported requests, completor abort, or completer timeout
16	0h RW/1C/V	**Correctable Error Detected (CED):** Indicates a correctable error was detected. Set when received an internal correctable error from receiver errors / framing errors, tlp crc error, dllp crc error, replay num rollover, replay timeout.
15	0h RO	**Reserved (RSVD_1):** Reserved
14:12	0h RO	**Max Read Request Size (MRRS):** Hardwired to 0. This field applies only to the PCIe link interface.
11	0h RO	**Enable No Snoop (ENS):** Not supported. The root port will never issue non-snoop requests.
10	0h RW/P	**Aux Power PM Enable (APME):** The OS will set this bit to '1' if the device connected has detected aux power.
9	0h RO	**Phantom Functions Enable (PFE):** Not supported
8	0h RO	**Extended Tag Field Enable (ETFE):** Not supported

Bit Range	Default & Access	Field Name (ID): Description
7:5	0h RW	**Max Payload Size (MPS):** The root port only supports up to 256B max payload. Programming this field to any values other than 128B or 256B max payload size will result in aliasing to 128B max payload size. If the DCAP.MPS indicates 128B max payload size support, programming this field to any values other than 128B will result in aliasing to 128B max payload size. Programming this field to any values greater than DCAP.MPS will result in aliasing to 128B max payload size. 000b: 128 bytes max payload size. 001b: 256 bytes max payload size. 010b: 512 bytes max payload size. 011b: 1024 bytes max payload size. 100b: 2048 bytes max payload size. 101b: 4096 bytes max payload size. 110b: Reserved. 111b: Reserved. This field applies only to the PCIe link interface. Note: Software should ensure that the system is quiescent and no TLP is in progress prior to changing this field. BIOS should program this field prior to enabling BME.
4	0h RO	**Enable Relaxed Ordering (ERO):** Not supported
3	0h RW	**Unsupported Request Reporting Enable (URE):** When set, allows signaling ERR_NONFATAL, ERR_FATAL, or ERR_COR to the Root Control register when detecting an unmasked Unsupported Request (UR). An ERR_COR is signaled when a unmasked Advisory Non-Fatal UR is received. An ERR_FATAL, ERR_or NONFATAL, is sent to the Root Control Register when an uncorrectable non-Advisory UR is received with the severity set by the Uncorrectable Error Severity register.
2	0h RW	**Fatal Error Reporting Enable (FEE):** enables signaling of ERR_FATAL to the Root Control register due to internally detected errors or error messages received across the link. Other bits also control the full scope of related error reporting.
1	0h RW	**Non-Fatal Error Reporting Enable (NFE):** When set, enables signaling of ERR_NONFATAL to the Root Control register due to internally detected errors or error messages received across the link. Other bits also control the full scope of related error reporting.
0	0h RW	**Correctable Error Reporting Enable (CEE):** When set, enables signaling of ERR_CORR to the Root Control register due to internally detected errors or error messages received across the link. Other bits also control the full scope of related error reporting.

18.16 Link Capabilities (LCAP) — Offset 4Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 4Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO/V	**Port Number (PN):** Indicates the port number for the root port. This value is different for each implemented port: Port # Value of PN field 1 01h 2 02h 3 03h 4 04h
23	0h RO	**Reserved (RSVD):** Reserved
22	0h RW/O	**ASPM Optionality Compliance (ASPMOC):** ASPM Optionality Compliance(ASPMOC): This bit must be set to 1b for PCIe 3.0 compliant port. Components implemented against certain earlier versions of this specification will have this bit set to 0b. Software is permitted to use the value of this bit to help determine whether to enable ASPM or whether to run ASPM compliance tests.
21	0h RO	**Link Bandwidth Notification Capability (LBNC):** This port supports Link Bandwidth Notification status and interrupt mechanisms.
20	0h RO	**Link Active Reporting Capable (LARC):** This port supports the optional capability of reporting the DL_Active state of the Data Link Control and Management State Machine.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RO	**Surprise Down Error Reporting Capable (SDERC):** Set to '0' to indicate the root port does not support Surprise Down Error Reporting
18	0h RO	**Clock Power Management (CPM):** '0' Indicates that root ports do not support the CLKREQ# mechanism.
17:15	0h RW/O	**L1 Exit Latency (EL1):** Indicates an exit latency of 2us to 4us. 000b Less than 1 us 001b 1 us to less than 2 us 010b 2 us to less than 4 us 011b 4 us to less than 8 us 100b 8 us to less than 16 us 101b 16 us to less than 32 us 110b 32 us to 64 us 111b More than 64 us Note: If PXP PLL shutdown is enabled, BIOS should program this latency to comprehend PLL lock latency.
14:12	0h RO/V	**L0s Exit Latency (EL0):** Indicates an exit latency based upon common-clock configuration: LCTL.CCC Value 0 MPC.UCEL 1 MPC.CCEL
11:10	0h RW/O	**Active State Link PM Support (APMS):** Indicates the level of active state power management on this link Bits Definition 00 No ASPM Supported 01 L0s Supported 10 L1 Supported 11 L0s and L1 supported Note: If STRPFUSECFG.ASPMDIS is 1, the default of this field is '01'. Otherwise, the default of this field is '11'. If STRPFUSECFG.ASPMDIS is 1, BIOS writing '11' to this field will have the same effect as writing '01'. '01' will be reflected on this register when read and the register will turn to Read-Only once written once.
9:4	0h RO/V	**Maximum Link Width (MLW):** For the root ports, several values can be taken, based upon the value of the chipset configuration register field RPC.PC1 for ports 1-4: Port # Value of PN field RPC.PC1 00 01 10 11 1 01h 02h 02h 04h 2 01h 01h 01h 01h 3 01h 01h 02h 01h 4 01h 01h 01h 01h
3:0	0h RO/V	**Max Link Speeds (MLS):** Indicates the supported link speeds of the Root Port. 0001b 2.5 GT/s Link speed supported 0010b 5.0 GT/s and 2.5GT/s Link speeds supported This register reports a value of 0001b if the Root Port Gen2 Disable Fuse is set or the MPC.PCIEGEN2DIS bit is set, else this register reports a value of 0010b. Max Link Speeds (MLS): This field indicates the maximum Link speed of the associated Port. The encoded value specifies a bit location in the Supported Link Speeds Vector (in the Link Capabilities 2 register) that corresponds to the maximum Link speed. Defined encodings are: 0001b: Supported Link Speeds Vector field bit 0. 0010b: Supported Link Speeds Vector field bit 1. 0011b: Supported Link Speeds Vector field bit 2. 0100b: Supported Link Speeds Vector field bit 3. 0101b: Supported Link Speeds Vector field bit 4. 0110b: Supported Link Speeds Vector field bit 5. 0111b: Supported Link Speeds Vector field bit 6. All other encodings are reserved. This register reports a value of 0001b if GEN1 data rate is supported but both GEN2 and GEN3 data rate support are disabled through Fuse or MPC.PCIESD register. This register reports a value of 0010b if both GEN1 and GEN2 data rate are supported but GEN3 data rate support is disabled through Fuse or MPC.PCIESD register.

18.17 Link Control; Link Status (LCTL_LSTS) — Offset 50h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 50h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C/V	**Link Autonomous Bandwidth Status (LABS):** This bit is Set by hardware to indicate that hardware has autonomously changed Link speed or width, without the Port transitioning through DL_Down status, for reasons other than to attempt to correct unreliable Link operation. This bit must be set if the Physical Layer reports a speed or width change was initiated by the Downstream component that was indicated as an autonomous change. The default value of this bit is 0b.

Bit Range	Default & Access	Field Name (ID): Description
30	0h RW/1C/V	**Link Bandwidth Management Status (LBMS):** This bit is Set by hardware to indicate that either of the following has occurred without the Port transitioning through DL_Down status: - A Link retraining has completed following a write of 1b to the Retrain Link bit Note: This bit is Set following any write of 1b to the Retrain Link bit, including when the Link is in the process of retraining for some other reason. - Hardware has changed Link speed or width to attempt to correct unreliable Link operation, either through an LTSSM timeout or a higher level process This bit must be set if the Physical Layer reports a speed or width change was initiated by the Downstream component that was not indicated as an autonomous change. The default value of this bit is 0b.
29	0h RO/V	**Link Active (LA):** Set to 1b when the Data Link Control and Management State Machine is in the DL_Active state, 0b otherwise.
28	0h RO/V	**Slot Clock Configuration (SCC):** In normal mode, root port uses the same reference clock as on the platform and does not generate its own clock. Note: The default of this register bit is dependent on the 'PCIe Non-Common Clock With SSC Mode Enable Strap'. If the strap enables non-common clock with SSC support, this bit shall default to '0'. Otherwise, this bit shall default to '1'.
27	0h RO/V	**Link Training (LT):** The root port sets this bit whenever link training is occurring, or that 1b was written to the Retrain Link bit but Link training has not yet begun. It clears the bit upon completion of link training.
26	0h RO	**Reserved (RSVD):** Reserved
25:20	0h RO/V	**Negotiated Link Width (NLW):** For the root ports, this register could take on several values: Port # Value of PN field RPC.PC1 00 01 10 11 1 01h 02h 02h 04h 2 01h 01h 01h 01h 3 01h 01h 02h 01h 4 01h 01h 01h 01h The value of this register is undefined if the link has not successfully trained.
19:16	0h RO/V	**Current Link Speed (CLS):** 0001b Link is 2.5Gb/s Link 0010b 5.0 GT/s Link This field indicates the negotiated Link speed of the given link. The encoded value specifies a bit location in the Supported Link Speeds Vector (in the Link Capabilities 2 register) that corresponds to the current Link speed. Defined encodings are: 0001b: Supported Link Speeds Vector field bit 0. 0010b: Supported Link Speeds Vector field bit 1. 0011b: Supported Link Speeds Vector field bit 2. 0100b: Supported Link Speeds Vector field bit 3. 0101b: Supported Link Speeds Vector field bit 4. 0110b: Supported Link Speeds Vector field bit 5. 0111b: Supported Link Speeds Vector field bit 6. All other encodings are reserved. The value of this field is undefined if the link is not up.
15:12	0h RO	**Reserved (RSVD_1):** Reserved
11	0h RW	**Link Autonomous Bandwidth Interrupt Enable (LABIE):** Link Autonomous Bandwidth Interrupt Enable - When Set, this bit enables the generation of an interrupt to indicate that the Link Autonomous Bandwidth Status bit has been Set.
10	0h RW	**Link Bandwidth Management Interrupt Enable (LBMIE):** When Set, this bit enables the generation of an interrupt to indicate that the Link Bandwidth Management Status bit has been Set. This bit is not applicable and is reserved for Endpoints, PCI Express-to-PCI/PCI-X bridges, and Upstream Ports of Switches. Functions that do not implement the Link Bandwidth Notification Capability must hardwire this bit to 0b. Default value of this bit is 0b.
9	0h RW	**Hardware Autonomous Width Disable (HAWD):** When Set, this bit disables hardware from changing the Link width for reasons other than attempting to correct unreliable Link operation by reducing Link width. Default value of this bit is 0b. Note: When operating as PCI Express, this bit defines the value of the Link Upconfigure Capability in TS2 Ordered Sets.
8	0h RO	**Enable Clock Power Management (ECPM):** Reserved. Not supported on Root Ports.
7	0h RW	**Extended Synch (ES):** When set, forces extended transmission of FTS ordered sets in FTS and extra TS2 at exit from L1 prior to entering L0. Note: This functionality is not applicable for Mobile Express.
6	0h RW	**Common Clock Configuration (CCC):** When set, indicates that the root port and device are operating with a distributed common reference clock.

Bit Range	Default & Access	Field Name (ID): Description
5	0h WO	**Retrain Link (RL):** When set, the root port will train its downstream link. This bit always returns '0' when read. Software uses LSTS.LT to check the status of training. It is permitted to write 1b to this bit while simultaneously writing modified values to other fields in this register. If the LTSSM is not already in Recovery or Configuration, the resulting Link training must use the modified values. If the LTSSM is already in Recovery or Configuration, the modified values are not required to affect the Link training that's already in progress.
4	0h RW	**Link Disable (LD):** When set, the root port will disable the link by directing the LTSSM to the Disabled state.
3	0h RW/O	**Read Completion Boundary Control (RCBC):** Indicates the read completion boundary is 64 bytes.
2	0h RO	**Reserved (RSVD_2):** Reserved
1:0	0h RW	**Active State Link PM Control (ASPM):** Indicates whether the root port should enter L0s or L1 or both. Bits Definition 00 Disabled 01 L0s Entry Enabled 10 L1 Entry Enabled 11 L0s and L1 Entry Enabled The value of this register is used unless the Root Port ASPM Control Override Enable register is set, in which case the Root Port ASPM Control Override value is used. Note: If STRPFUSECFG.ASPMDIS is '1', hardware will always see '00' as an output from this register. BIOS reading this register should always return the correct value.

18.18 Slot Capabilities (SLCAP) — Offset 54h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 54h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW/O	**Physical Slot Number (PSN__31_24):** This is a value that is unique to the slot number. BIOS sets this field and it remains set until a platform reset.
23:19	0h RW/O	**Physical Slot Number (PSN__23_19):** This is a value that is unique to the slot number. BIOS sets this field and it remains set until a platform reset.
18	0h RO	**No Command Completed Support (NCCS):** Set to '1' as this port does not implement a Hot Plug controller and can handle back-2-back writes to all fields of the slot control register without delay between successive writes.
17	0h RO	**Electromechanical Interlock Present (EMIP):** Set to 0 to indicate that no electro-mechanical interlock is implemented.
16:15	0h RW/O	**Slot Power Limit Scale (SLS):** specifies the scale used for the slot power limit value. BIOS sets this field and it remains set until a platform reset.
14:8	0h RW/O	**Slot Power Limit Value (SLV__14_8):** Specifies the upper limit (in conjunction with SLS value), on the upper limit on power supplied by the slot. The two values together indicate the amount of power in watts allowed for the slot. BIOS sets this field and it remains set until a platform reset.
7	0h RW/O	**Slot Power Limit Value (SLV__7_7):** Specifies the upper limit (in conjunction with SLS value), on the upper limit on power supplied by the slot. The two values together indicate the amount of power in watts allowed for the slot. BIOS sets this field and it remains set until a platform reset.
6	0h RW/O	**Hot Plug Capable (HPC):** When set, Indicates that hot plug is supported.
5	0h RW/O	**Hot Plug Surprise (HPS):** When set, indicates the device may be removed from the slot without prior notification.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RO	**Power Indicator Present (PIP):** Indicates that a power indicator LED is not present for this slot.
3	0h RO	**Attention Indicator Present (AIP):** Indicates that an attention indicator LED is not present for this slot.
2	0h RO	**MRL Sensor Present (MSP):** Indicates that an MRL sensor is not present
1	0h RO	**Power Controller Present (PCP):** Indicates that a power controller is not implemented for this slot
0	0h RO	**Attention Button Present (ABP):** Indicates that an attention button is not implemented for this slot.

18.19 Slot Control; Slot Status (SLCTL_SLSTS) — Offset 58h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 58h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD):** Reserved
24	0h RW/1C/V	**Data Link Layer State Changed (DLLSC):** This bit is set when the value reported in Data Link Layer Link Active field of the Link Status register is changed. In response to a Data Link Layer State Changed event, software must read Data Link Layer Link Active field of the Link Status register to determine if the link is active before initiating configuration cycles to the hot plugged device.
23	0h RO	**Electromechanical Interlock Status (EMIS):** Reserved as this port does not support and electromechanical interlock.
22	0h RO/V	**Presence Detect State (PDS):** If XCAP.SI is set (indicating that this root port spawns a slot), then this bit indicates whether a device is connected ('1') or empty ('0'). If XCAP.SI is cleared, this bit is a '1'.
21	0h RO	**MRL Sensor State (MS):** Reserved as the MRL sensor is not implemented.
20	0h RO	**Command Completed (CC):** This register is RO as this port does not implement a Hot Plug Controller..
19	0h RW/1C/V	**Presence Detect Changed (PDC):** This bit is set by the root port when the SLSTS.PDS bit changes state.
18	0h RO	**MRL Sensor Changed (MSC):** Reserved as the MRL sensor is not implemented.
17	0h RO	**Power Fault Detected (PFD):** Reserved as a power controller is not implemented.
16	0h RO	**Attention Button Pressed (ABP):** This register is RO as this port does not implement an attention button
15:13	0h RO	**Reserved (RSVD_1):** Reserved
12	0h RW	**Data Link Layer State Changed Enable (DLLSCE):** When set, this field enables generation of a hot plug interrupt when the Data Link Layer Link Active field is changed

Bit Range	Default & Access	Field Name (ID): Description
11	0h RO	**Electromechanical Interlock Control (EMIC):** Reserved as this port does not support an Electromechanical Interlock.
10	0h RO	**Power Controller Control (PCC):** This bit has no meaning for module based hot plug.
9:8	0h RO	**Power Indicator Control (PIC):** This register is RO as this port does not implement a Hot Plug Controller..
7:6	0h RO	**Attention Indicator Control (AIC):** This register is RO as this port does not implement a Hot Plug Controller..
5	0h RW	**Hot Plug Interrupt Enable (HPE):** When set, enables generation of a hot plug interrupt on enabled hot plug events.
4	0h RO	**Command Completed Interrupt Enable (CCE):** This register is RO as this port does not implement a Hot Plug Controller..
3	0h RW	**Presence Detect Changed Enable (PDE):** When set, enables the generation of a hot plug interrupt or wake message when the presence detect logic changes state.
2	0h RO	**MRL Sensor Changed Enable (MSE):** This register is RO as this port does not implement a Hot Plug Controller..
1	0h RO	**Power Fault Detected Enable (PFE):** This register is RO as this port does not implement a Hot Plug Controller...
0	0h RO	**Attention Button Pressed Enable (ABE):** This register is RO as this port does not implement a Hot Plug Controller..

18.20 Root Control (RCTL) — Offset 5Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 5Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:4	0h RO	**Reserved (RSVD_1):** Reserved
3	0h RW	**PME Interrupt Enable (PIE):** When set, enables interrupt generation when RSTS.PS is in a set state (either due to a '0' to '1' transition, or due to this bit being set with RSTS.PS already set).
2	0h RW	**System Error on Fatal Error Enable (SFE):** When set, an SERR# will be generated if a fatal error is reported by any of the devices in the hierarchy of this root port, including fatal errors in this root port. This register is not dependent on CMD.SEE being set.
1	0h RW	**System Error on Non-Fatal Error Enable (SNE):** When set, an SERR# will be generated if a non-fatal error is reported by any of the devices in the hierarchy of this root port, including non-fatal errors in this root port. This register is not dependent on CMD.SEE being set.
0	0h RW	**System Error on Correctable Error Enable (SCE):** When set, an SERR# will be generated if a correctable error is reported by any of the devices in the hierarchy of this root port, including correctable errors in this root port. This register is not dependent on CMD.SEE being set.

18.21 Root Status (RSTS) — Offset 60h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 60h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Reserved (RSVD):** Reserved
17	0h RO/V	**PME Pending (PP):** Indicates another PME is pending when the PME status bit is set. When the original PME is cleared by software, it will be set again, the requestor ID will be updated, and this bit will be cleared. Root ports have a one deep PME pending queue.
16	0h RW/1C/V	**PME Status (PS):** Indicates that PME was asserted by the requestor ID in RID. Subsequent PMEs are kept pending until this bit is cleared.
15:0	0h RO/V	**PME Requestor ID (RID):** Indicates the PCI requestor ID of the last PME requestor. Valid only when PS is set. Root ports are capable of storing the requester ID for two PM_PME messages, with one active (this register) and a one deep pending queue. Subsequent PM_PME messages will be dropped.

18.22 Device Capabilities 2 (DCAP2) — Offset 64h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 64h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RO	**Reserved (RSVD):** Reserved
19:18	0h RW/O	**Optimized Buffer Flush/Fill Supported (OBFFS):** 00b - OBFF is not supported. 01b - OBFF is supported using Message signaling only. 10b - OBFF is supported using WAKE# signaling only. 11b - OBFF is supported using WAKE# and Message signaling. BIOS should program this field to 00b or 10b during system initialization to advertise the level of hardware OBFF support to software. BIOS should never program this field to 01b or 11b since OBFF messaging is not supported.
17:12	0h RO	**Reserved (RSVD_1):** Reserved
11	0h RW/O	**LTR Mechanism Supported (LTRMS):** A value of 1b indicates support for the optional Latency Tolerance Reporting (LTR) mechanism capability. BIOS must write to this register with either a '1' or a '0' to enable/disable the root port from declaring support for the LTR capability.
10:6	0h RO	**Reserved (RSVD_2):** Reserved
5	0h RO	**ARI Forwarding Supported (AFS):** ARI Forwarding Supported (AFS): Applicable only to Switch Downstream Ports and Root Ports; must be 0b for other Function types. This bit must be set to 1b if a Switch Downstream Port or Root Port supports this optional capability. Note: This bit is not made RWO to simplify implementation, since there is a requirement that the ARI Forwarding Enable bit must be hardwired to 0b if ARI Forwarding Supported bit is 0b. It is low risk to keep this risk 1b.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RO	**Completion Timeout Disable Supported (CTDS):** A value of 1b indicates support for the Completion Timeout Disable mechanism.
3:0	0h RO	**Completion Timeout Ranges Supported (CTRS):** This field indicates device support for the optional Completion Timeout programmability mechanism. This mechanism allows system software to modify the Completion Timeout value. This field is applicable only to Root Ports, Endpoints that issue requests on their own behalf, and PCI Express to PCI/PCI-X Bridges that take ownership of requests issued on PCI Express. For all other devices this field is reserved and must be hardwired to 0000b. Four time value ranges are defined: Range A: 50us to 10ms Range B: 10ms to 250ms Range C: 250ms to 4s Range D: 4s to 64s Bits are set according to show timeout value ranges supported. 0000b Completion Timeout programming not supported. 0001b Range A 0010b Range B 0011b Ranges A [amp] B 0110b Ranges B [amp] C 0111b Ranges A, B [amp] C [lt]-- This is what PCH supports 1110b Ranges B, C [amp] D 1111b Ranges A, B, C [amp] D All other values are reserved.

18.23 Device Control 2; Device Status 2 (DCTL2_DSTS2) — Offset 68h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 68h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15	0h RO	**Reserved (RSVD_1):** Reserved
14:13	0h RW	**Optimized Buffer Flush/Fill Enable (OBFFEN):** 00b Disable OBFF mechanism. 01b Enable OBFF mechanism using Message signaling (Variation A). 10b Enable OBFF mechanism using Message signaling (Variation B). 11b Enable OBFF using WAKE# signaling. Note: Only encoding 00b and 11b are supported. The encoding of 01b or 10b would be aliased to 00b. If DCAP2.OBFFS is clear, programming this field to any non-zero values will have no effect.
12:11	0h RO	**Reserved (RSVD_2):** Reserved
10	0h RW	**LTR Mechanism Enable (LTREN):** When Set to 1b, this bit enables the Latency Tolerance Reporting (LTR) mechanism. For Downstream Ports, this bit must be reset to the default value if the Port goes to DL_Down status. If DCAP2.LTRMS is clear, programming this field to any non-zero values will have no effect.
9:6	0h RO	**Reserved (RSVD_3):** Reserved
5	0h RW	**ARI Forwarding Enable (AFE):** ARI Forwarding Enable (AFE): When set, the Downstream Port disables its traditional Device Number field being 0b enforcement when turning a Type 1 Configuration Request into a Type 0 Configuration Request, permitting access to Extended Functions in an ARI Device immediately below the Port.
4	0h RW	**Completion Timeout Disable (CTD):** When set to 1b, this bit disables the Completion Timeout mechanism. This field is required for all devices that support the Completion Timeout Disable Capability. Software is permitted to set or clear this bit at any time. When set, the Completion Timeout detection mechanism is disabled. If there are outstanding requests when the bit is cleared, it is permitted but not required for hardware to apply the completion timeout mechanism to the outstanding requests. If this is done, it is permitted to base the start time for each request on either the time this bit was cleared or the time each request was issued.

Bit Range	Default & Access	Field Name (ID): Description
3:0	0h RW	**Completion Timeout Value (CTV):** In Devices that support Completion Timeout programmability, this field allows system software to modify the Completion Timeout value. This field is applicable to Root Ports, Endpoints that issue requests on their own behalf, and PCI Express to PCI/PCI-X Bridges that take ownership of requests issued on PCI Express. For all other devices this field is reserved and must be hardwired to 0000b. A Device that does not support this optional capability must hardwire this field to 0000b and is required to implement a timeout value in the range 50us to 50ms. Devices that support Completion Timeout programmability must support the values given below corresponding to the programmability ranges indicated in the Completion Timeout Values Supported field. The root port targeted configurable ranges are listed below, along with the range allowed by the PCI Express 2.0 specification. Defined encodings: 0000b Default range: 40-50ms (spec range 50us to 50ms) Values available if Range A (50us to 10 ms) programmability range is supported: 0001b 90-100us (spec range is 50 us to 100 us) 0010b 9-10ms (spec range is 1ms to 10 ms) Values available if Range B (10ms to 250ms) programmability range is supported: 0101b 40-50ms (spec range is 16ms to 55ms) 0110b 160-170ms (spec range is 65ms to 210ms) Values available if Range C (250ms to 4s) programmability range is supported: 1001b 400-500ms (spec range is 260ms to 900ms) 1010b 1.6-1.7s (spec range is 1s to 3.5s) Values not defined above are Reserved. Software is permitted to change the value in this field at any time. For requests already pending when the Completion Timeout Value is changed, hardware is permitted to use either the new or the old value for the outstanding requests, and is permitted to base the start time for each request either when this value was changed or when each request was issued.

18.24 Link Capabilities 2 (LCAP2) — Offset 6Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 6Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RO	**Reserved (RSVD):** Reserved
22:16	0h RO	**Lower SKP OS Reception Supported Speeds Vector (LSOSRSS):** Lower SKP OS Reception Supported Speeds Vector(LSOSRSS): If this field is non-zero, it indicates that the Port, when operating at the indicated speed(s) supports SRIS and also supports receiving SKP OS at the rate defined for SRNS while running in SRIS. Bit definitions within this field are: Bit 0 2.5 GT/s Bit 1 5.0 GT/s Bit 2 8.0 GT/s Bits 6:3 RsvdP Behavior is undefined if a bit is set in this field and the corresponding bit is not set in the Supported Link Speeds Vector.
15:9	0h RO	**Lower SKP OS Generation Supported Speeds Vector (LSOSGSSV):** Lower SKP OS Generation Supported Speeds Vector(LSOSGSSV): If this field is non-zero, it indicates that the Port, when operating at the indicated speed(s) supports SRIS and also supports software control of the SKP Ordered Set transmission scheduling rate. Bit definitions within this field are: Bit 0 2.5 GT/s Bit 1 5.0 GT/s Bit 2 8.0 GT/s Bits 6:3 RsvdP Behavior is undefined if a bit is set in this field and the corresponding bit is not set in the Supported Link Speeds Vector.
8	0h RO	**Crosslink Supported (CS):** Not supported.
7:1	0h RO/V	**Supported Link Speeds Vector (SLSV):** Supported Link Speeds Vector (SLSV): This field indicates the supported Link speed of the associated Port. For each bit, a value of 1b indicates that the corresponding Link speed is supported; otherwise, the Link speed is not supported. Bit definitions within this field are: Bit 0: 2.5 GT/s. Bit 1: 5.0 GT/s. Bit 2: 8.0 GT/s. Bits 6:3: Reserved. This register reports a value of 0001b if GEN1 data rate is supported but both GEN2 and GEN3 data rate support are disabled through Fuse or MPC.PCIESD register. This register reports a value of 0011b if both GEN1 and GEN2 data rate are supported but GEN3 data rate support is disabled through Fuse or MPC.PCIESD register.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**Reserved (RSVD_1):** Reserved

18.25 Link Control 2; Link Status 2 (LCTL2_LSTS2) — Offset 70h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 70h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (RSVD):** Reserved
21	0h RW/1C/V/P	**Link Equalization Request (LER):** Link Equalization Request (LER): This bit is set by hardware to request the Link equalization process to be performed on the Link. Register Attribute: Dynamic.
20	0h RO/V/P	**Equalization Phase 3 Successful (EQP3S):** Equalization Phase 3 Successful (EQP3S): When set to 1, this bit indicates that Phase 3 of the Transmitter Equalization procedure has successfully completed.
19	0h RO/V/P	**Equalization Phase 2 Successful (EQP2S):** Equalization Phase 2 Successful (EQP2S): When set to 1, this bit indicates that Phase 2 of the Transmitter Equalization procedure has successfully completed.
18	0h RO/V/P	**Equalization Phase 1 Successful (EQP1S):** Equalization Phase 1 Successful (EQP1S): When set to 1, this bit indicates that Phase 1 of the Transmitter Equalization procedure has successfully completed.
17	0h RO/V/P	**Equalization Complete (EQC):** Equalization Complete (EC): When set to 1, this bit indicates that the Transmitter Equalization procedure has completed
16	0h RO/V	**Current De-emphasis Level (CDL):** When the Link is operating at 5.0 GT/s speed, this bit reflects the level of de-emphasis. Encodings: 1b -3.5 dB 0b -6 dB The value in this bit is undefined when the Link is not operating at 5.0 GT/s speed.
15:12	0h RW/P	**Compliance Preset/De-emphasis (CD):** For 8.0 GT/s Data Rate: This field sets the Transmitter Preset in Polling.Compliance state if the entry occurred due to the Enter Compliance bit being 1b. Results are undefined if a reserved preset encoding is used when entering Polling.Compliance in this way. For 5.0 GT/s Data Rate: This bit sets the de-emphasis level in Polling.Compliance state if the entry occurred due to the Enter Compliance bit being 1b. Encodings: 0001b -3.5 dB 0000b -6 dB When the Link is not operating at 5.0 GT/s speed, the setting of this bit has no effect. The default value of this field is 0000b. This bit is intended for debug, compliance testing purposes. System firmware and software is allowed to modify this bit only during debug or compliance testing. In all other cases, the system must ensure that this field is set to the default value.
11	0h RW/P	**Compliance SOS (CSOS):** When set to 1b, the LTSSM is required to send SKP Ordered Sets periodically in between the (modified) compliance patterns. The default value of this bit is 0b. This bit is applicable when the Link is operating at 2.5 GT/s or 5.0 GT/s data rates only.
10	0h RW/P	**Enter Modified Compliance (EMC):** When this bit is set to 1b, the device transmits Modified Compliance Pattern if the LTSSM enters Polling.Compliance substate. Default value of this bit is 0b. This register is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this register only during debug or compliance testing. In all other cases, the system must ensure that this register is set to the default value.

Bit Range	Default & Access	Field Name (ID): Description
9:7	0h RW/P	**Transmit Margin (TM):** This field controls the value of the nondeemphasized voltage level at the Transmitter pins. This field is reset to 000b on entry to the LTSSM Polling.Configuration substate (see PCI Express Chapter 4 for details of how the Transmitter voltage level is determined in various states). Encodings: 000b Normal operating range 001b 800-1200 mV for full swing and 400-700 mV for half-swing 010b - (n-1) Values must be monotonic with a non-zero slope. The value of n must be greater than 3 and less than 7. At least two of these must be below the normal operating range of n : 200-400 mV for full-swing and 100-200 mV for half-swing n - 111b reserved For a Multi-Function device associated with an Upstream Port, the field in Function 0 is of type RWS, and only Function 0 controls the component's Link behavior. In all other Functions of that device, this field is of type RsvdP. Default value of this field is 000b. Components that support only the 2.5 GT/s speed are permitted to hardwire this bit to 000b. This register is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this register only during debug or compliance testing. In all other cases, the system must ensure that this register is set to the default value.
6	0h RW/P	**Selectable De-emphasis (SD):** When the Link is operating at 5.0 GT/s speed, this bit selects the level of de-emphasis for an Upstream component. Encodings: 1b -3.5 dB 0b -6 dB When the Link is not operating at 5.0 GT/s speed, the setting of this bit has no effect. When the Link is operating at 2.5 GT/s speed, the setting of this bit has no effect.
5	0h RO	**Hardware Autonomous Speed Disable (HASD):** Reserved. This port cannot autonomously change speeds.
4	0h RW/P	**Enter Compliance (EC):** Software is permitted to force a Link to enter Compliance mode at the speed indicated in the Target Link Speed field by setting this bit to 1b in both components on a Link and then initiating a hot reset on the Link. Default value of this bit following Fundamental Reset is 0b. This bit is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this bit only during debug or compliance testing. In all other cases, the system must ensure that this bit is set to the default value. This bit is intended for debug, compliance testing purposes only. System firmware and software is allowed to modify this bit only during debug or compliance testing. In all other cases, the system must ensure that this bit is set to the default value.
3:0	0h RW/V/P	**Target Link Speed (TLS):** Target Link Speed (TLS): This field sets an upper limit on Link operational speed by restricting the values advertised by the upstream component in its training sequences. The encoded value specifies a bit location in the Supported Link Speeds Vector (in the Link Capabilities 2 register) that corresponds to the current Link speed. Defined encodings are: 0001b: Supported Link Speeds Vector field bit 0. 0010b: Supported Link Speeds Vector field bit 1. 0011b: Supported Link Speeds Vector field bit 2. 0100b: Supported Link Speeds Vector field bit 3. 0101b: Supported Link Speeds Vector field bit 4. 0110b: Supported Link Speeds Vector field bit 5. 0111b: Supported Link Speeds Vector field bit 6. All other encodings are reserved. If a value is written to this field that does not correspond to a supported speed, as indicated by the Supported Link Speeds Vector, the result is undefined. The default value of this field is GEN1. Note: This register field could be used by REUT software to limit the link speed to 2.5 GT/s or 5 GT/s data rate.

18.26 Slot Capabilities 2 (SLCAP2) — Offset 74h

Size:32 bits

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 74h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD):** Reserved

18.27 Slot Control 2; Slot Status 2 (SLCTL2_SLSTS2) — Offset 78h

Size:32 bits

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 78h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:0	0h RO	**Reserved (RSVD_1):** Reserved

18.28 Message Signaled Interrupt Identifiers; Message Signaled Interrupt Message Control (MID_MC) — Offset 80h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 80h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23	0h RO	**64-Bit Address Capable (C64):** Capable of generating a 32-bit message only.
22:20	0h RW	**Multiple Message Enable (MME):** These bits are RW for software compatibility, but only one message is ever sent by the root port.
19:17	0h RO	**Multiple Message Capable (MMC):** Only one message is required.
16	0h RW	**MSI Enable (MSIE):** If set, MSI is enabled and traditional interrupt pins are not used to generate interrupts. CMD.BME must be set for an MSI to be generated. If CMD.BME is cleared, and this bit is set, no interrupts (not even pin based) are generated.
15:8	0h RW/O	**NEXT:** Indicates the location of the next capability in the list. The default value of this register is 90h which points to the Subsystem Vendor capability structure. BIOS can determine which capabilities will be exposed by including or removing them from the capability linked list. As this register is RWO, BIOS must write a value to this register, even if it is to re-write the default value.
7:0	0h RO	**Capability ID (CID):** Capabilities ID indicates MSI.

18.29 Message Signaled Interrupt Message Address (MA) — Offset 84h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 84h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**ADDR:** Lower 32 bits of the system specified message address, always DW aligned.
1:0	0h RO	**Reserved (RSVD):** Reserved

18.30 Message Signaled Interrupt Message Data (MD) — Offset 88h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 88h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:0	0h RW	**DATA:** This 16-bit field is programmed by system software if MSI is enabled. Its content is driven onto the lower word (PCI AD[lb]15:0[rb]) during the data phase of the MSI memory write transaction.

18.31 Subsystem Vendor Capability (SVCAP) — Offset 90h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 90h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:8	0h RW/O	**NEXT:** Indicates the location of the next capability in the list. The default value of this register is A0h which points to the PCI Power Management capability structure. BIOS can determine which capabilities will be exposed by including or removing them from the capability linked list. As this register is RWO, BIOS must write a value to this register, even if it is to re-write the default value.
7:0	0h RO	**Capability Identifier (CID):** Value of 0Dh indicates this is a PCI bridge subsystem vendor capability.

18.32 Subsystem Vendor IDs (SVID) — Offset 94h

Size:32 bits

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 94h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**Subsystem Identifier (SID):** Indicates the subsystem as identified by the vendor. This field is write once and is locked down until a bridge reset occurs (not the PCI bus reset).
15:0	0h RW/O	**Subsystem Vendor Identifier (SVID):** Indicates the manufacturer of the subsystem. This field is write once and is locked down until a bridge reset occurs (not the PCI bus reset).

18.33 Power Management Capability; PCI Power Management Capabilities (PMCAP_PMC) — Offset A0h

Size:32 bits

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + A0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	0h RO	**PME Support (PMES):** Indicates PME# is supported for states D0, D3HOT and D3COLD. The root port does not generate PME#, but reporting that it does is necessary for legacy Microsoft* operating systems to enable PME# in devices connected behind this root port.
26	0h RO	**D2_Support (D2S):** The D2 state is not supported.
25	0h RO	**D1_Support (D1S):** The D1 state is not supported.
24:22	0h RO	**Aux_Current (AC):** Reports 0mA (self-powered), as use of this controller does not add to suspect well power consumption.
21	0h RO	**Device Specific Initialization (DSI):** Indicates that no device-specific initialization is required.
20	0h RO	**Reserved (RSVD):** Reserved
19	0h RO	**PME Clock (PMEC):** Indicates that PCI clock is not required to generate PME#.
18:16	0h RO	**Version (VS):** Indicates support for Revision 1.2 of the PCI Power Management Specification.
15:8	0h RO	**NEXT:** Indicates this is the last item in the list.

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Capability Identifier (CID):** Value of 01h indicates this is a PCI power management capability.

18.34 PCI Power Management Control And Status (PMCS) — Offset A4h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + A4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Data (DTA):** Reserved
23	0h RO	**Bus Power / Clock Control Enable (BPCE):** Reserved per PCI Express specification
22	0h RO	**B2/B3 Support (B23S):** Reserved per PCI Express specification.
21:16	0h RO	**Reserved (RSVD):** Reserved
15	0h RO	**PME Status (PMES):** Indicates a PME was received on the downstream link.
14:13	0h RO	**Data Scale (DSC):** Reserved
12:9	0h RO	**Data Select (DSEL):** Reserved
8	0h RW/P	**PME Enable (PMEE):** Indicates PME is enabled. The root port takes no action on this bit, but it must be RW for legacy Microsoft* operating systems to enable PME# on devices connected to this root port.
7:4	0h RO	**Reserved (RSVD_1):** Reserved
3	0h RW/O	**No Soft Reset (NSR):** When set to 1 this bit indicates that devices transitioning from D3hot to D0 because of Power State commands do not perform an internal reset. Configuration context is preserved. Upon transition from D3hot to D0 Initialized state, no additional operating system intervention is required to preserve Configuration Context beyond writing the Power State bits. When clear, devices do perform an internal reset upon transitioning from D3hot to D0 via software control of the Power State bits. Configuration Context is lost when performing the soft reset. Upon transition from D3hot to D0 state, full reinitialization sequence is needed to return the device to D0 Initialized. Regardless of this bit, devices that transition from D3hot to D0 by a system or bus segment reset will return to the device state D0 Uninitialized with only PME context preserved if PME is supported and enabled.
2	0h RO	**Reserved (RSVD_2):** Reserved
1:0	0h RW	**Power State (PS):** This field is used both to determine the current power state of the root port and to set a new power state. The values are: 00 D0 state 11 D3HOT state When in the D3HOT state, the controller's configuration space is available, but the I/O and memory spaces are not. Type 1 configuration cycles are also not accepted. Interrupts are not required to be blocked as software will disable interrupts prior to placing the port into D3HOT. If software attempts to write a '10' or '01' to these bits, the write will be ignored.

18.35 Advanced Error Extended Reporting Capability Header (AECH) — Offset 100h

Size:32 bits The AER capability can optionally be included or excluded from the capabilities list. The full AER is supported.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 100h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW/O	**Next Capability Offset (NCO):** Set to 000h as this is the last capability in the list.
19:16	0h RW/O	**Capability Version (CV):** For systems that support AER, BIOS should write a 1h to this register else it should write 0
15:0	0h RW/O	**Capability ID (CID):** For systems that support AER, BIOS should write a 0001h to this register else it should write 0

18.36 Uncorrectable Error Status (UES) — Offset 104h

This register must maintain its state through a platform reset. It loses its state upon loss of core power.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 104h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (RSVD):** Reserved
21	0h RW/1C/V/P	**ACS Violation Status (AVS):** Reserved. Access Control Services are not supported
20	0h RW/1C/V/P	**Unsupported Request Error Status (URE):** Indicates an unsupported request was received.
19	0h RO	**ECRC Error Status (EE):** ECRC is not supported.
18	0h RW/1C/V/P	**Malformed TLP Status (MT):** Indicates a malformed TLP was received.
17	0h RW/1C/V/P	**Receiver Overflow Status (RO):** Indicates a receiver overflow occurred.
16	0h RW/1C/V/P	**Unexpected Completion Status (UC):** Indicates an unexpected completion was received.
15	0h RW/1C/V/P	**Completor Abort Status (CA):** Indicates a completer abort was received

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW/1C/V/P	**Completion Timeout Status (CT):** Indicates a completion timed out. This is signaled if Completion Timeout is enabled and a completion fails to return within the amount of time specified by the Completion Timeout Value
13	0h RO	**Flow Control Protocol Error Status (FCPE):** Not supported.
12	0h RW/1C/V/P	**Poisoned TLP Status (PT):** Indicates a poisoned TLP was received.
11:6	0h RO	**Reserved (RSVD_1):** Reserved
5	0h RO	**Surprise Down Error Status (SDE):** Surprise Down is not supported.
4	0h RW/1C/V/P	**Data Link Protocol Error Status (DLPE):** Indicates a data link protocol error occurred.
3:1	0h RO	**Reserved (RSVD_2):** Reserved
0	0h RO	**Training Error Status (TE):** Not supported.

18.37 Uncorrectable Error Mask (UEM) — Offset 108h

When set, the corresponding error in the UES register is masked, and the logged error will cause no action. When cleared, the corresponding error is enabled. This register is only reset by a loss of core power.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 108h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (RSVD):** Reserved
21	0h RW/P	**ACS Violation Mask (AVM):** Reserved. Access Control Services are not supported
20	0h RW/P	**Unsupported Request Error Mask (URE):** Mask for uncorrectable errors.
19	0h RO	**ECRC Error Mask (EE):** ECRC is not supported.
18	0h RW/P	**Malformed TLP Mask (MT):** Mask for malformed TLPs
17	0h RW/P	**Receiver Overflow Mask (RO):** Mask for receiver overflows.
16	0h RW/P	**Unexpected Completion Mask (UC):** Mask for unexpected completions.
15	0h RW/P	**Completor Abort Mask (CM):** Mask for completer abort.

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW/P	**Completion Timeout Mask (CT):** Mask for completion timeouts.
13	0h RO	**Flow Control Protocol Error Mask (FCPE):** Not supported.
12	0h RW/P	**Poisoned TLP Mask (PT):** Mask for poisoned TLPs.
11:6	0h RO	**Reserved (RSVD_1):** Reserved
5	0h RO	**Surprise Down Error Mask (SDE):** Surprise Down is not supported.
4	0h RW/P	**Data Link Protocol Error Mask (DLPE):** Mask for data link protocol errors.
3:1	0h RO	**Reserved (RSVD_2):** Reserved
0	0h RO	**Training Error Mask (TE):** Not supported.

18.38 Uncorrectable Error Severity (UEV) — Offset 10Ch

This register gives the option to make an uncorrectable error fatal or non-fatal. An error is fatal if the bit is set. An error is non-fatal if the bit is cleared. This register is only reset by a loss of core power.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 10Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (RSVD):** Reserved
21	0h RW/P	**ACS Violation Severity (AVS):** Severity for ACS violation.
20	0h RW/P	**Unsupported Request Error Severity (URE):** Severity for unsupported request reception.
19	0h RO	**ECRC Error Severity (EE):** ECRC is not supported.
18	0h RW/P	**Malformed TLP Severity (MT):** Severity for malformed TLP reception.
17	0h RW/P	**Receiver Overflow Severity (RO):** Severity for receiver overflow occurrences.
16	0h RW/P	**Unexpected Completion Severity (UC):** Severity for unexpected completion reception.
15	0h RW/P	**Completor Abort Severity (CA):** Severity for completer abort.
14	0h RW/P	**Completion Timeout Severity (CT):** Severity for completion timeout.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RO	**Flow Control Protocol Error Severity (FCPE):** Not supported.
12	0h RW/P	**Poisoned TLP Severity (PT):** Severity for poisoned TLP reception.
11:6	0h RO	**Reserved (RSVD_1):** Reserved
5	0h RO	**Surprise Down Error Severity (SDE):** Surprise Down is not supported.
4	0h RW/P	**Data Link Protocol Error Severity (DLPE):** Severity for data link protocol errors.
3:1	0h RO	**Reserved (RSVD_2):** Reserved
0	0h RO	**Training Error Severity (TE):** TE not supported. This bit is left as RO='1' for ease of implementation..

18.39 Correctable Error Status (CES) — Offset 110h

This register is only reset by a loss of core power

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 110h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**Reserved (RSVD):** Reserved
13	0h RW/1C/V/P	**Advisory Non-Fatal Error Status (ANFES):** When set, indicates that an Advisory Non-Fatal Error occurred.
12	0h RW/1C/V/P	**Replay Timer Timeout Status (RTT):** Indicates the replay timer timed out.
11:9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RW/1C/V/P	**Replay Number Rollover Status (RNR):** Indicates the replay number rolled over.
7	0h RW/1C/V/P	**Bad DLLP Status (BD):** Indicates a bad DLLP was received.
6	0h RW/1C/V/P	**Bad TLP Status (BT):** Indicates a bad TLP was received.
5:1	0h RO	**Reserved (RSVD_2):** Reserved
0	0h RW/1C/V/P	**RE:** Indicates a receiver error occurred.

18.40 Correctable Error Mask (CEM) — Offset 114h

When set, the corresponding error in the CES register is masked, and the logged error will cause no action. When cleared, the corresponding error is enabled. This register is only reset by a loss of core power.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 114h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**Reserved (RSVD):** Reserved
13	0h RW/P	**Advisory Non-Fatal Error Mask (ANFEM):** When set, masks Advisory Non-Fatal errors from (a) signaling ERR_COR to the device control register and (b) updating the Uncorrectable Error Status register. This register is set by default to enable compatibility with software that does not comprehend Role-Based Error Reporting.
12	0h RW/P	**Replay Timer Timeout Mask (RTT):** Mask for replay timer timeout.
11:9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RW/P	**Replay Number Rollover Mask (RNR):** Mask for replay number rollover.
7	0h RW/P	**Bad DLLP Mask (BD):** Mask for bad DLLP reception.
6	0h RW/P	**Bad TLP Mask (BT):** Mask for bad TLP reception.
5:1	0h RO	**Reserved (RSVD_2):** Reserved
0	0h RW/P	**RE:** Mask for receiver errors.

18.41 Advanced Error Capabilities and Control (AECC) — Offset 118h

This register is only reset by a loss of core power.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 118h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Reserved (RSVD):** Reserved
8	0h RO	**ECRC Check Enable (ECE):** ECRC is not supported.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**ECRC Check Capable (ECC):** ECRC is not supported.
6	0h RO	**ECRC Generation Enable (EGE):** ECRC is not supported.
5	0h RO	**ECRC Generation Capabile (EGC):** ECRC is not supported.
4:0	0h RO/V/P	**First Error Pointer (FEP):** Identifies the bit position of the first error reported in the Uncorrectable Error Status Register.

18.42 Header Log DW1 (HL_DW1) — Offset 11Ch

Size:32 bits These registers report the header for the TLP corresponding to a detected error. This register is only reset by a loss of core power.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 11Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V/P	**1st dWord of TLP (DW1):** Byte0 [amp][amp] Byte1 [amp][amp] Byte2 [amp][amp] Byte3

18.43 Header Log DW2 (HL_DW2) — Offset 120h

Size:32 bits These registers report the header for the TLP corresponding to a detected error. This register is only reset by a loss of core power.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 120h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V/P	**2nd dWord of TLP (DW2):** Byte4 [amp][amp] Byte5 [amp][amp] Byte6 [amp][amp] Byte7

18.44 Header Log DW3 (HL_DW3) — Offset 124h

Size:32 bits These registers report the header for the TLP corresponding to a detected error. This register is only reset by a loss of core power.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 124h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V/P	**3rd dWord of TLP (DW3):** Byte8 [amp][amp] Byte9 [amp][amp] Byte10 [amp][amp] Byte11

18.45 Header Log DW4 (HL_DW4) — Offset 128h

Size:32 bits These registers report the header for the TLP corresponding to a detected error. This register is only reset by a loss of core power.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 128h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V/P	**4th dWord of TLP (DW4):** Byte12 [amp][amp] Byte13 [amp][amp] Byte14 [amp][amp] Byte15

18.46 Root Error Command (REC) — Offset 12Ch

This register allows errors to generate interrupts.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 12Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RO	**Reserved (RSVD):** Reserved
2	0h RW	**Fatal Error Reporting Enable (FERE):** When set, the root port will generate an interrupt when a fatal error is reported by the attached device.
1	0h RW	**Non-fatal Error Reporting Enable (NERE):** When set, the root port will generate an interrupt when a non-fatal error is reported by the attached device.
0	0h RW	**Correctable Error Reporting Enable (CERE):** When set, the root port will generate an interrupt when a correctable error is reported by the attached device.

18.47 Root Error Status (RES) — Offset 130h

This register can track more than one error and set the 'multiple' bits if a second or subsequent error occurs and the first has not been serviced. This register is only reset by a loss of core power.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 130h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	0h RO	**Advanced Error Interrupt Message Number (AEMN):** Reserved. There is only one error interrupt allocated.
26:7	0h RO	**Reserved (RSVD):** Reserved
6	0h RW/1C/V/P	**Fatal Error Message Received (FEMR):** Set when one or more Fatal Uncorrectable Error Messages have been received.
5	0h RW/1C/V/P	**Non-Fatal Error Messages Received (NFEMR):** Set when one or more Non-Fatal Uncorrectable error messages have been received
4	0h RW/1C/V/P	**First Uncorrectable Fatal (FUF):** Set when the first Uncorrectable Error message received is for a fatal error.
3	0h RW/1C/V/P	**Multiple ERR_FATAL/NONFATAL Received (MENR):** Set when either a fatal or a non-fatal error is received and the ENR bit is already set.
2	0h RW/1C/V/P	**ERR_FATAL/NONFATAL Received (ENR):** Set when either a fatal or a non-fatal error message is received.
1	0h RW/1C/V/P	**Multiple ERR_COR Received (MCR):** Set when a correctable error message is received and the CR bit is already set.
0	0h RW/1C/V/P	**ERR_COR Received (CR):** Set when a correctable error message is received.

18.48 Error Source Identification (ESID) — Offset 134h

Size:32 bits Identifies the source (Requester ID) of the first correctable and uncorrectable (Non-Fatal / Fatal) errors reported in the Root Error Status register. This register is only reset by a loss of core power.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 134h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO/V/P	**ERR_FATAL/NONFATAL Source Identification (EFNFSID):** Loaded with the Requester ID indicated in the received ERR_FATAL or ERR_NONFATAL Message with the ERR_FATAL/NONFATAL Received register is not already set.
15:0	0h RO/V/P	**ERR_COR Source Identification (ECSID):** Loaded with the Requester ID indicated in the received ERR_COR Message with the ERR_COR Received register is not already set.

18.49 ACS Extended Capability Header (ACSECH) — Offset 140h

Size:32 bits

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 140h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW/O	**Next Capability Offset (NCO):** Next Capability Offset (NCO): Points to the next capability.
19:16	0h RW/O	**Capability Version (CV):** Capability Version (CV): For systems that support ACS Extended Capability, BIOS should write a 1h to this register else it should write 0.
15:0	0h RW/O	**Capability ID (CID):** Capability ID (CID): For systems that support ACS Extended Capability, BIOS should write a 000Dh to this register else it should write 0.

18.50 ACS Capability Register; ACS Control Register (ACSCAPR_ACSCTLR) — Offset 144h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 144h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RO	**Reserved (RSVD):** Reserved
22	0h RO	**ACS Direct Translated P2P Enable (TE):** ACS Direct Translated P2P Enable (TE): ACS Direct Translated P2P is not supported.
21	0h RO	**ACS P2P Egress Control Enable (EE):** ACS P2P Egress Control Enable (EE): ACS P2P Egress Control is not supported.
20	0h RO	**ACS Upstream Forwarding Enable (UE):** ACS Upstream Forwarding Enable (UE): ACS Upstream Forwarding is not supported.
19	0h RW	**ACS P2P Completion Redirect Enable (CE):** ACS P2P Completion Redirect Enable (CE): Determines when the component redirects peer-to-peer Completions upstream; applicable only to Read Completions whose Relaxed Ordering Attribute is clear.
18	0h RW	**ACS P2P Request Redirect Enable (RE):** ACS P2P Request Redirect Enable (RE): Determines when the component redirects peer-to-peer memory Requests targeting another peer port upstream. I/O, Configuration, VDM Messages and Completions are never affected by ACS P2P Request Redirect.
17	0h RW	**ACS Translation Blocking Enable (BE):** ACS Translation Blocking Enable (BE): When set, the component blocks all upstream Memory Requests whose Address Translation (AT) field is not set to the default value. I/O, Configuration, Completions and Messages are never affected by ACS Translation Blocking.
16	0h RW	**ACS Source Validation Enable (VE):** ACS Source Validation Enable (VE): When set, the component validates the Bus Number from the Requester ID of upstream Requests against the secondary / subordinate Bus Numbers. I/O, Configuration and Completions are never affected by ACS Source Validation.

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	**Reserved (RSVD_1):** Reserved
7	0h RO	**Reserved (RSVD_2):** Reserved
6	0h RO	**ACS Direct Translated P2P (T):** ACS Direct Translated P2P (T): ACS Direct Translated P2P is not supported.
5	0h RO	**ACS P2P Egress Control (E):** ACS P2P Egress Control (E): ACS P2P Egress Control is not supported.
4	0h RO	**ACS Upstream Forwarding (U):** ACS Upstream Forwarding (U): ACS Upstream Forwarding is not supported.
3	0h RW/O	**ACS P2P Completion Redirect (C):** ACS P2P Completion Redirect (C): Required for all Functions that support ACS P2P Request Redirect; must be hardwired to 0b otherwise. If 1b, indicates that the component implements ACS P2P Completion Redirect.
2	0h RW/O	**ACS P2P Request Redirect (R):** ACS P2P Request Redirect (R): Required for Root Ports that support peer-to-peer traffic with other Root Ports; required for Switch Downstream Ports; required for multi-function device Functions that support peer-to-peer traffic with other Functions; must be hardwired to 0b otherwise. If 1b, indicates that the component implements ACS P2P Request Redirect.
1	0h RW/O	**ACS Translation Blocking (B):** ACS Translation Blocking (B): Required for Root Ports and Switch Downstream Ports; must be hardwired to 0b otherwise. If 1b, indicates that the component implements ACS Translation Blocking.
0	0h RW/O	**ACS Source Validation (V):** ACS Source Validation (V): Required for Root Ports and Switch Downstream Ports; must be hardwired to 0b otherwise. If 1b, indicates that the component implements ACS Source Validation.

18.51 PTM Extended Capability Header (PTMECH) — Offset 150h

Size:32 bits

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 150h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW/O	**Next Capability Offset (NCO):** Next Capability Offset (NCO): Points to the next capability.
19:16	0h RW/O	**Capability Version (CV):** Capability Version (CV): For systems that support PTM Extended Capability, BIOS should write a 1h to this register else it should write 0.
15:0	0h RW/O	**Capability ID (CID):** Capability ID (CID): For systems that support PTM Extended Capability, BIOS should write a 001Fh to this register else it should write 0.

18.52 PTM Capability Register (PTMCAPR) — Offset 154h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 154h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:8	0h RW/O	**Local Clock Granularity (LCG):** Local Clock Granularity(LCG): 0000 0000b - Time Source does not implement a local clock. It simply propagates timing information obtained from further Upstream in the PTM Hierarchy when responding to PTM Request messages. 0000 0001b - 1111 1110b: Indicates the period of this Time Sources local clock in ns. 1111 1111b: Indicates the period of this Time Sources local clock is greater than 254 ns. If the PTM Root Select bit is Set, this local clock is used to provide PTM Master Time. Otherwise, the Time Source uses this local clock to locally track PTM Master Time received from further Upstream within a PTM Hierarchy.
7:3	0h RO	**Reserved (RSVD_1):** Reserved
2	0h RW/O	**PTM Root Capable (PTMRC):** PTM Root Capable(PTMRC): Root Ports must set this bit to 1b.
1	0h RW/O	**PTM Responder Capable (PTMRSPC):** PTM Responder Capable(PTMRSPC): Root Ports are permitted to set this bit to 1b to indicate that they implement the PTM Responder role.
0	0h RO	**PTM Requester Capable (PTMREQC):** PTM Requester Capable(PTMREQC): PTM Requester Role is not supported by Root Port.

18.53 PTM Control Register (PTMCTLR) — Offset 158h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 158h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:8	0h RO	**Effective Granularity (EG):** Effective Granularity(EG): Root Port does not support PTM Requester role.
7:2	0h RO	**Reserved (RSVD_1):** Reserved
1	0h RW	**Root Select (RS):** Root Select(RS): When Set, if the PTM Enable bit is also Set, this Time Source is the PTM Root. Within each PTM Hierarchy, it is recommended that system software select only the furthest Upstream Time Source to be the PTM Root.
0	0h RW	**PTM Enable (PTME):** PTM Enable(PTME): When Set, this Function is permitted to participate in the PTM mechanism according to its selected role. Software must not have the PTM Enable bit Set in the PTM Control register on a Function associated with an Upstream Port unless the associated Downstream Port on the Link already has the PTM Enable bit Set in its associated PTM Control register. Register Attribute: Static.

18.54 L1 Sub-States Extended Capability Header (L1SECH) — Offset 200h

Size:32 bits Note: When operating in Mobile Express mode, this capability should not be enabled.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 200h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW/O	**Next Capability Offset (NCO):** This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are Reserved and must be implemented as 00b and software must mask them to allow for future uses of these bits.
19:16	0h RW/O	**Capability Version (CV):** This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. Must be 1h for this version of the specification. For systems that support L1 Sub-State Extended Capability, BIOS should set this field to 1h
15:0	0h RW/O	**PCI Express Extended Capability ID (PCIEEC):** This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. For systems that support L1 Sub-State Extended Capability, BIOS should set this field to 001Eh. .

18.55 L1 Sub-States Capabilities (L1SCAP) — Offset 204h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 204h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23:19	0h RW/O	**Port Tpower_on Value (PTV):** Along with the Port T_POWER_ON Scale Field in the L1 Substates Capabilities register sets theTime (in us) that this Port requires the port on the opposite side of Link to wait in L1.OFF_EXIT after sampling CLKREQ# asserted before actively driving the interface. Port Tpower_on is calculated by multiplying the value in this field by the value in the Port Tpower_on scale field in the L1 Sub-States Capabilities 2 register. Required for all Ports that support L1.OFF.
18	0h RO	**Reserved (RSVD_1):** Reserved
17:16	0h RW/O	**Port Tpower_on Scale (PTPOS):** Specifies the scale used for Tpower_on value field in the L1 Substates Capabilities register. '00b': 2 us '01b': 10 us '10b': 100 us '11b': Reserved Required for all Ports that support L1.OFF.
15:8	0h RW/O	**Port Common Mode Restore Time (PCMRT):** This is the time (in us) required for this Port to re-establish common mode. Required for all ports that support L1.OFF.
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4	0h RW/O	**L1 PM Substates Supported (L1PSS):** When Set this bit indicates that this Port supports L1 PM Substates. For compatibility with possible future extensions, software must not enable L1 PM Substates unless this bit is set. This RWO field must be programmed prior to enabling ASPM.
3	0h RW/O	**ASPM L1.1 Substates Supported (AL11S):** When set, this bit indicates that this port supports L1 substates for ASPM L1.SNOOZ. Required for both Upstream and Downstream ports. This RWO field must be programmed prior to enabling ASPM. Register Attribute: Static

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW/O	**ASPM L1.2 Supported (AL12S):** When set, this bit indicates that ASPM_L1.OFF is supported. Required for both Upstream and Downstream ports. This RWO field must be programmed prior to enabling ASPM. Register Attribute: Static
1	0h RW/O	**PCI-PM L1.1 Supported (PPL11S):** When set, this bit indicates that L1.SNOOZ sub-state is supported and this bit must be set by all ports implementing L1 Sub-States. A port that supports L1.OFF must support L1.SNOOZ. Required for both upstream and downstream ports. This RWO field must be programmed prior to enabling ASPM. Register Attribute: Static
0	0h RW/O	**PCI-PM L1.2 Supported (PPL12S):** When set, this bit indicates that L1.OFF power management feature is supported. Required for both upstream and downstream ports. This RWO field must be programmed prior to enabling ASPM. Register Attribute: Static

18.56 L1 Sub-States Control 1 (L1SCTL1) — Offset 208h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 208h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RW	**L1.2 LTR Threshold Latency ScaleValue (L12LTRTLSV):** This field contains the L1.OFF LTR Threshold Latency Scale Value for this particular PCIe root port. The value in this field, together with L12LTRTLV is compared against both the snoop and non-snoop LTR values of the device. 000: L12LTRSTLV times 1 ns 001: L12LTRSTLV times 32 ns 010: L12LTRSTLV times 1024 ns 011: L12LTRSTLV times 32768 ns 100: L12LTRSTLV times 1048576 ns 101: L12LTRSTLV times 33554432 ns Others: Not Permitted. This field must be programmed prior to enabling L1.OFF. Register Attribute: Static
28:26	0h RO	**Reserved (RSVD):** Reserved
25:16	0h RW	**L1.2 LTR Threshold Latency Value (L12OFFLTRTLV):** This field contains the L1.2 LTR Threshold Latency Value for this particular PCIe root port. The value in this field, together with L12LTRTLSV is compared against both the snoop and non-snoop LTR values of the device. This field must be programmed prior to enabling L1.OFF. Register Attribute: Static
15:8	0h RW	**Common Mode Restore Time (CMRT):** This is the Tcommon_mode time the PCIe root port needs to continue sending TS1 and refrain from sending TS2 in Recovery state to allow the TX common mode to be established prior to sending TS2. The timer starts from the time when the first TS1 has been sent and the receiver has detected un-squelch. The value in this field defines the time in micro-seconds. This field must be programmed prior to enabling L1.OFF. Register Attribute: Static
7:4	0h RO	**Reserved (RSVD_1):** Reserved
3	0h RW	**ASPM L1.1 Enabled (AL11E):** When set, this bit indicates that ASPM L1.SNOOZ substates are enabled for ASPM. Required for both upstream and downstream ports. Register Attribute: Dynamic
2	0h RW	**ASPM L1.2 Enable (AL12E):** When set, this bit indicates that ASPM L1.OFF substates are enabled for PCI-PM. Required for both upstream and downstream ports. Register Attribute: Dynamic
1	0h RW	**PCI-PM L1.SNOOZ Enable (PPL11E):** When set, this bit indicates that PCI-PM L1.SNOOZ power management feature is enabled. If L1.OFF is enabled, L1.SNOOZ must also be enabled. This field must be programmed prior to enabling ASPM L1. Register Attribute: Dynamic Note: If STRPFUSECFG.mPHYIOPMDIS is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**PCI-PM L1.2 Enabled (PPL12E):** When set, this bit indicates that PCI-PM L1.OFF power management feature is enabled. L1.OFF can only be enabled if the platform supports bi-directional CLKREQPLUS#. This field must be programmed prior to enabling ASPM L1. Ports that support L1.OFF shall support Latency Tolerance Reporting. Register Attribute: Dynamic Note: If STRPFUSECFG.mPHYIOPMDIS is '1', hardware will always see '0' as an output from this register. BIOS reading this register should always return the correct value.

18.57 L1 Sub-States Control 2 (L1SCTL2) — Offset 20Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 20Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD):** Reserved
7:3	0h RW	**Power On Wait Time (POWT):** Along with the Tpower_on Scale sets the minimum amount of time (in us) that the Port must wait in L1.OFF EXIT after sampling CLKREQPLUS# asserted before actively driving the interface. The timer starts counting when CLKREQPLUS# is sampled asserts in L1.OFF state. Tpower_on value is calculated by multiplying the value in this field by the value in the TPOS field. This field must be programmed prior to enabling L1.OFF. Register Attribute: Static
2	0h RO	**Reserved (RSVD_1):** Reserved
1:0	0h RW	**Tpower_on Scale (TPOS):** Specifies the scale used for Tpower_on value. '00b': 2 us '01b': 10 us '10b': 100us '11b': Reserved. Required for all Ports that support L1.OFF. Register Attribute: Static

18.58 Secondary PCI Express Extended Capability Header (SPEECH) — Offset 220h

Size:32 bits Note: When operating in Mobile Express mode, this capability should not be enabled.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 220h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW/O	**Next Capability Offset (NCO):** Next Capability Offset (NCO): This field contains the offset to the next PCI Express Capability structure or 000h if no other items exist in the linked list of Capabilities. For Extended Capabilities implemented in Configuration Space, this offset is relative to the beginning of PCI compatible Configuration Space and thus must always be either 000h (for terminating list of Capabilities) or greater than 0FFh. The bottom 2 bits of this offset are Reserved and must be implemented as 00b and software must mask them to allow for future uses of these bits.

Bit Range	Default & Access	Field Name (ID): Description
19:16	0h RW/O	**Capability Version (CV):** Capability Version (CV): This field is a PCI-SIG defined version number that indicates the version of the Capability structure present. For systems that support Secondary PCI Express Extended Capability, BIOS should write a 1h to this register else it should write 0
15:0	0h RW/O	**PCI Express Extended Capability ID (PCIEECID):** PCI Express Extended Capability ID (PCIEECID): This field is a PCI-SIG defined ID number that indicates the nature and format of the Extended Capability. PCI Express Extended Capability ID for the Secondary PCI Express Extended Capability is 0019h. For systems that support Secondary PCI Express Extended Capability, BIOS should write a 0019h to this register else it should write 0.

18.59 Link Control 3 (LCTL3) — Offset 224h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 224h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:9	0h RO	**Enable Lower SKP OS Generation Vector (ELSOSGV):** Enable Lower SKP OS Generation Vector(ELSOSGV): When the Link is in L0 and the bit in this field corresponding to the current Link speed is Set, SKP Ordered Sets are scheduled at the rate defined for SRNS, overriding the rate required based on the clock tolerance architecture. Bit definitions within this field are: Bit 0 2.5 GT/s Bit 1 5.0 GT/s Bit 2 8.0 GT/s Bits 6:3 RsvdP Behavior is undefined if a bit is Set in this field and the corresponding bit in the Lower SKP OS Generation Supported Speeds Vector is not set.
8:2	0h RO	**Reserved (RSVD_1):** Reserved
1	0h RW	**Link Equalization Request Interrupt Enable (LERIE):** Link Equalization Request Interrupt Enable (LERIE): When set, this bit enables the generation of an interrupt to indicate that the Link Equalization Request bit has been set.
0	0h RW	**PE:** Perform Equalization (PE): When this bit is 1b and Link Retrain bit is set with the Target Link Speed field set to 8 GT/s, the Downstream Port must perform Link Equalization. This bit is cleared by Root Port upon entry to Link Equalization

18.60 Lane Error Status (LES) — Offset 228h

The Lane number is the default Lane number which is invariant to Link width and Lane reversal negotiation that occurs during training. Based on the Port Configuration setting, if the maximum link width of the port is less than x4, the upper lane status bits should be ignored.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 228h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RSVD):** Reserved
3	0h RW/1C/V/P	**Lane 3 Error Status (L3ES):** Lane 3 Error Status (L3ES): Lane 3 detected a Lane-based error.
2	0h RW/1C/V/P	**Lane 2 Error Status (L2ES):** Lane 2 Error Status (L2ES): Lane 2 detected a Lane-based error.
1	0h RW/1C/V/P	**Lane 1 Error Status (L1ES):** Lane 1 Error Status (L1ES): Lane 1 detected a Lane-based error.
0	0h RW/1C/V/P	**Lane 0 Error Status (L0ES):** Lane 0 Error Status (L0ES): Lane 0 detected a Lane-based error.

18.61 Lane 0 and Lane 1 Equalization Control (L01EC) — Offset 22Ch

Each entry contains the values for the Lane with the corresponding default Lane number which is invariant to Link width and Lane reversal negotiation that occurs during Link training. Based on the Port Configuration setting, if the maximum link width of the port is less than x4, the upper lane status bits should be ignored

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 22Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved
30:28	0h RW	**Upstream Port Lane 1 Receiver Preset Hint (UPL1RPH):** Upstream Port Lane 1 Receiver Preset Hint (UPL1RPH): Field contains the Receiver Preset Hint value sent or received during Link Equalization.
27:24	0h RW	**Upstream Port Lane 1 Transmitter Preset (UPL1TP):** Upstream Port Lane 1 Transmitter Preset (UPL1TP): Field contains the Transmit Preset value sent or received during Link Equalization.
23	0h RO	**Reserved (RSVD_1):** Reserved
22:20	0h RW	**Downstream Port Lane 1 Receiver Preset Hint (DPL1RPH):** Downstream Port Lane 1 Receiver Preset Hint (DPL1RPH): Receiver Preset Hint may be used as a hint for receiver equalization by this Port when the Port is operating as a Downstream Port.
19:16	0h RW	**Downstream Port Lane 1 Transmitter Preset (DPL1TP):** Downstream Port Lane 1 Transmitter Preset (DPL1TP): Transmitter Preset used for equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port.
15	0h RO	**Reserved (RSVD_2):** Reserved
14:12	0h RW	**Upstream Port Lane 0 Receiver Preset Hint (UPL0RPH):** Upstream Port Lane 0 Receiver Preset Hint (UPL0RPH): Field contains the Receiver Preset Hint value sent or received during Link Equalization.
11:8	0h RW	**Upstream Port Lane 0 Transmitter Preset (UPL0TP):** Upstream Port Lane 0 Transmitter Preset (UPL0TP): Field contains the Transmit Preset value sent or received during Link Equalization.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**Reserved (RSVD_3):** Reserved
6:4	0h RW	**Downstream Port Lane 0 Receiver Preset Hint (DPL0RPH):** Downstream Port Lane 0 Receiver Preset Hint (DPL0RPH): Receiver Preset Hint may be used as a hint for receiver equalization by this Port when the Port is operating as a Downstream Port.
3:0	0h RW	**Downstream Port Lane 0 Transmitter Preset (DPL0TP):** Downstream Port Lane 0 Transmitter Preset (DPL0TP): Transmitter Preset used for equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port.

18.62 Lane 2 and Lane 3 Equalization Control (L23EC) — Offset 230h

Each entry contains the values for the Lane with the corresponding default Lane number which is invariant to Link width and Lane reversal negotiation that occurs during Link training. Based on the Port Configuration setting, if the maximum link width of the port is less than x4, the upper lane status bits should be ignored

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 230h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved
30:28	0h RW	**Upstream Port Lane 3 Receiver Preset Hint (UPL3RPH):** Upstream Port Lane 3 Receiver Preset Hint (UPL3RPH): Field contains the Receiver Preset Hint value sent or received during Link Equalization.
27:24	0h RW	**Upstream Port Lane 3 Transmitter Preset (UPL3TP):** Upstream Port Lane 3 Transmitter Preset (UPL3TP): Field contains the Transmit Preset value sent or received during Link Equalization.
23	0h RO	**Reserved (RSVD_1):** Reserved
22:20	0h RW	**Downstream Port Lane 3 Receiver Preset Hint (DPL3RPH):** Downstream Port Lane 3 Receiver Preset Hint (DPL3RPH): Receiver Preset Hint may be used as a hint for receiver equalization by this Port when the Port is operating as a Downstream Port.
19:16	0h RW	**Downstream Port Lane 3 Transmitter Preset (DPL3TP):** Downstream Port Lane 3 Transmitter Preset (DPL3TP): Transmitter Preset used for equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port.
15	0h RO	**Reserved (RSVD_2):** Reserved
14:12	0h RW	**Upstream Port Lane 2 Receiver Preset Hint (UPL2RPH):** Upstream Port Lane 2 Receiver Preset Hint (UPL2RPH): Field contains the Receiver Preset Hint value sent or received during Link Equalization.
11:8	0h RW	**Upstream Port Lane 2 Transmitter Preset (UPL2TP):** Upstream Port Lane 2 Transmitter Preset (UPL2TP): Field contains the Transmit Preset value sent or received during Link Equalization.
7	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
6:4	0h RW	**Downstream Port Lane 2 Receiver Preset Hint (DPL2RPH):** Downstream Port Lane 2 Receiver Preset Hint (DPL2RPH): Receiver Preset Hint may be used as a hint for receiver equalization by this Port when the Port is operating as a Downstream Port.
3:0	0h RW	**Downstream Port Lane 2 Transmitter Preset (DPL2TP):** Downstream Port Lane 2 Transmitter Preset (DPL2TP): Transmitter Preset used for equalization by this Port when the Port is operating as a Downstream Port. This field is ignored when the Port is operating as an Upstream Port.

18.63 PCI Express Replay Timer Policy 1 (PCIERTP1) — Offset 300h

The Replay Timer controlled by the Replay Timeout field is started when the Retry Buffer is empty and a TLP is placed into it or an Ack/Nak DLLP is received and there are still non-acknowledged packets within the Retry Buffer. The counter continues to count until the next valid Ack DLLP or a NAK DLLP that acknowledges unacknowledged TLPs is received, or it reaches the timeout value specified by this register. When a valid Ack/Nak DLLP is received, the timer is reset to zero and restarted if there are still non-acknowledged packets within the Retry Buffer. Otherwise if the Retry Buffer is empty, the counter is just reset to zero. If the timer reaches the timeout value, the non-acknowledged packets within the Retry Buffer will be replayed. The default for this register is dependant on the MAX_PAYLOAD_SIZE , the NEGOTIATED_WIDTH, and the NEGOTIATED_SPEED.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 300h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23:20	0h RW	**Gen 2 x1 (G2X1):** Gen 2 x1 (G2X1): Determines how many link clock cycles the Data Link Layer will wait to replay the contents of the Retry Buffer if a valid Ack/Nak DLLP is not received. The Replay Timeout value to be used varies based on the effective Maximum Payload Size. For 128B MPS: nnn * 64 link clocks. For 256B MPS: (nnn + 4) * 64 link clocks. For 512B MPS: (nnn + 7) * 64 link clocks. For PCIE Gen 2 speed and x1 width For Mobile Express HS-Gear 3 speed and x1 width.
19:16	0h RW	**Gen 2 x2 (G2X2):** Gen 2 x2 (G2X2): Determines how many link clock cycles the Data Link Layer will wait to replay the contents of the Retry Buffer if a valid Ack/Nak DLLP is not received. The Replay Timeout value to be used varies based on the effective Maximum Payload Size. For 128B MPS: nnn * 64 link clocks. For 256B MPS: (nnn + 2) * 64 link clocks. For 512B MPS: (nnn + 4) * 64 link clocks. For PCIe Gen 2 speed and x2 width. For Mobile Express HS-Gear 3 speed and x2 width.
15:12	0h RW	**Gen 2 x4 (G2X4):** Gen 2 x4 (G2X4): Determines how many link clock cycles the Data Link Layer will wait to replay the contents of the Retry Buffer if a valid Ack/Nak DLLP is not received. The Replay Timeout value to be used varies based on the effective Maximum Payload Size. For 128B MPS: nnn * 64 link clocks. For 256B MPS: (nnn + 2) * 64 link clocks. For 512B MPS: (nnn + 3) * 64 link clocks. For PCIe Gen 2 speed and x4 width. For Mobile Express HS-Gear 3 speed and x4 width.

Bit Range	Default & Access	Field Name (ID): Description
11:8	0h RW	**Gen 1 x1 (G1X1):** Gen 1 x1 (G1X1): Determines how many symbol times (i.e. number of link clock cycles) the Data Link Layer will wait to replay the contents of the Retry Buffer if a valid Ack/Nak DLLP is not received. The Replay Timeout value to be used varies based on the effective Maximum Payload Size. For 128B MPS: nnn * 64 link clocks. For 256B MPS: (nnn + 10) * 64 link clocks. For 512B MPS: (nnn + 17) * 64 link clocks. For PCIe Gen 1 speed and x1 width. For Mobile Express HS-Gear 2 speed and x1 width.
7:4	0h RW	**Gen 1 x2 (G1X2):** Gen 1 x2 (G1X2): Determines how many symbol times (i.e. number of link clock cycles) the Data Link Layer will wait to replay the contents of the Retry Buffer if a valid Ack/Nak DLLP is not received. The Replay Timeout value to be used varies based on the effective Maximum Payload Size. For 128B MPS: nnn * 64 link clocks. For 256B MPS: (nnn + 4) * 64 link clocks. For 512B MPS: (nnn + 8) * 64 link clocks. For PCIe Gen 1 speed and x2 width. For Mobile Express HS-Gear 2 speed and x2 width.
3:0	0h RW	**Gen 1 x4 (G1X4):** Gen 1 x4 (G1X4): Determines how many symbol times (i.e. number of link clock cycles) the Data Link Layer will wait to replay the contents of the Retry Buffer if a valid Ack/Nak DLLP is not received. The Replay Timeout value to be used varies based on the effective Maximum Payload Size. For 128B MPS: nnn * 64 link clocks. For 256B MPS: (nnn + 2) * 64 link clocks. For 512B MPS: (nnn + 3) * 64 link clocks. For PCIe Gen 1 speed and x4 width. For Mobile Express HS-Gear 2 speed and x4 width.

18.64 PCI Express Replay Timer Policy 2 (PCIERTP2) — Offset 304h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 304h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RW	**Lane 0 Lane Number (L0LN):** Lane 0 Lane Number(L0LN): This field specifies the lane number to be used in the per-lane GEN3 scrambling/descrambling logic for lane 0 when entering Loopback from Configuration state, prior to Link width negotiation and for GEN3 Polling Compliance entry where the lane number is not available. This field should never be used on all other cases, including entry to Loopback from L0.
29:28	0h RW	**Lane 1 Lane Number (L1LN):** Lane 1 Lane Number(L1LN): This field specifies the lane number to be used in the per-lane GEN3 scrambling/descrambling logic for lane 1 when entering Loopback from Configuration state, prior to Link width negotiation and for GEN3 Polling Compliance entry where the lane number is not available. This field should never be used on all other cases, including entry to Loopback from L0.
27:26	0h RW	**Lane 2 Lane Number (L2LN):** Lane 2 Lane Number(L2LN): This field specifies the lane number to be used in the per-lane GEN3 scrambling/descrambling logic for lane 2 when entering Loopback from Configuration state, prior to Link width negotiation and for GEN3 Polling Compliance entry where the lane number is not available. This field should never be used on all other cases, including entry to Loopback from L0.
25:24	0h RW	**Lane 3 Lane Number (L3LN):** Lane 3 Lane Number(L3LN): This field specifies the lane number to be used in the per-lane GEN3 scrambling/descrambling logic for lane 3 when entering Loopback from Configuration state, prior to Link width negotiation and for GEN3 Polling Compliance entry where the lane number is not available. This field should never be used on all other cases, including entry to Loopback from L0.
23	0h RW	**Loopback Master EQ TS1 Enable (LMEQTS1E):** Loopback Master EQ TS1 Enable(LMEQTS1E): When set, the Loopback Master will use EQ TS1 Ordered Sets to direct the Loopback Slave into Loopback from Configuration.Linkwidth.Start. The Preset field of the EQ TS1 Ordered Sets will be specified by Upstream Port Lane X Transmitter Preset and Upstream Port Lane X Receiver Preset Hint fields in the Lane Equalization Control registers.

Bit Range	Default & Access	Field Name (ID): Description
22	0h RW	**Loopback Master EQ Change Enable (LMEQCE):** Loopback Master EQ Change Enable(LMEQCE): This field is applicable to the case where Loopback is entered from Recovery state. When set, the Loopback Master will set the EC field of the GEN3 TS1 Ordered Sets to the appropriate value based on the ports direction(10b or 11b) to direct the Loopback Slave into Loopback from Recovery state. The Preset field of the GEN3 TS1 Ordered Sets will be specified by Upstream Port Lane X Transmitter Preset and Upstream Port Lane X Receiver Preset Hint fields in the Lane Equalization Control registers.
21:12	0h RO	**Reserved (RSVD):** Reserved
11:8	0h RW	**Gen 3 x1 (G3X1):** Gen 3 x1 (G3X1): Determines how many link clock cycles the Data Link Layer will wait to replay the contents of the Retry Buffer if a valid Ack/Nak DLLP is not received. The Replay Timeout value to be used varies based on the effective Maximum Payload Size. For 128B MPS: nnn * 64 link clocks. For 256B MPS: (nnn + 4) * 64 link clocks. For 512B MPS: (nnn + 8) * 64 link clocks. For Gen 3 speed and x1 width
7:4	0h RW	**Gen 3 x2 (G3X2):** Gen 3 x2 (G3X2): Determines how many link clock cycles the Data Link Layer will wait to replay the contents of the Retry Buffer if a valid Ack/Nak DLLP is not received. The Replay Timeout value to be used varies based on the effective Maximum Payload Size. For 128B MPS: nnn * 64 link clocks. For 256B MPS: (nnn + 2) * 64 link clocks. For 512B MPS: (nnn + 3) * 64 link clocks. For Gen 3 speed and x2 width
3:0	0h RW	**Gen 3 x4 (G3X4):** Gen 3 x4 (G3X4): Determines how many link clock cycles the Data Link Layer will wait to replay the contents of the Retry Buffer if a valid Ack/Nak DLLP is not received. The Replay Timeout value to be used varies based on the effective Maximum Payload Size. For 128B MPS: nnn * 64 link clocks. For 256B MPS: (nnn + 1) * 64 link clocks. For 512B MPS: (nnn + 2) * 64 link clocks. For Gen 3 speed and x4 width

18.65 PCI Express Status 1 (PCIESTS1) — Offset 328h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 328h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO/V	**LTSM State (LTSMSTATE):** Indicates the LTSM present state. Hex LTSSM States 00DETIDLE 01DETRDY 02DETIDLEP1TOP2 03DETRDYP2TOP1 04DETRDYINP1 05DETRDYINP1EXE 06DETP2POLLSTART 07DETP1POLLSTART 08DETP2TOP0 09DETP1TOP0 0AECPCMPRETRAIN 0BECPCMFAILP0TOP2 0CDETP0TOP2 0DPMODECHANGE 0EECPCMPCIE 0FECPCMUSB3 10DET2POLLINP0 11POLLINGACTIVE 12POLLINGCOMPLIANCEMARGINCNT 13POLLINGCOMPLIANCE 14POLLINGCOMPLIANCESPEEDUP 15POLLINGCOMPLIANCESPEEDDN 16POLLINGCOMPLIANCESPEEDTXEIDLE 17POLLINGCOMPLIANCESPEEDRXEIDLE 18POLLINGCOMPLIANCESPEED 19POLLINGCOMPLIANCESPEEDDONE 1APOLLINGCOMPLIANCEEXIT 1BPOLLINGCONFIGURATION 1CPOLLINGTXEIDLE 1DPOLLINGEND 1EPOLLINGENDWAIT 1FLINKWIDTHSTART 20LINKWIDTHACCEPT 21LANENUMWAIT 22LANENUMACCEPT 23LANEDESKEW 24CONFIGCOMPLETE 25CONFIGIDLE 26LWNEXITRECOVERY 27CONFIGLPBKENTRY 28CONFIGLPBKLWSTART 29CONFIGLPBKSPEEDTXEIDLE 2ACONFIGLPBKSPEEDSTART 2BCONFIGLPBKSPEEDRXEIDLE 2CCONFIGLPBKSPEED 2DCONFIGLPBKREUTSKIP 2ECONFIGLPBKREUT 2FCONFIGLPBKEXITM 30CONFIGLPBKTXEIDLE 31LWNEXIT 32LWNLNK2DETECT 33L0 34TXL0SRXL0 35RXL0STXL0 36TXL0SRXL0S 37L1TXEIDLE 38L1RCVEIDLE 39L1PREENTRY 3AL1ENTRY 3BL1IDLE 3CL1IDLEGEN2WAIT 3DL1EXIT 3EL2TXEIDLE 3FL2RCVEIDLE 40L2IDLEWAIT 41L2IDLERDY 42L2IDLE 43LOOPBACKENTRY 44LPBKACTIVEMTXSKP 45LPBKACTIVEMSKPDSKW 46LOOPBACKACTIVEM 47LPBKSLAVESPEEDTXEIDLE 48LPBKSLAVESPEEDRXEIDLE 49LPBKSLAVESPEED 4ALOOPBACKACTIVES 4BLOOPBACKCMMSKP 4CLOOPBACKCMM 4DLOOPBACKEXITM 4ELOOPBACKEXITS 4FLOOPBACKEXITL0 50LOOPBACKLNK2DETECT 51LOOPBACK2DETECT 52DISTX16TS1DIS 53DISTXEIDLE 54DISWAITSTART 55DISWAITGNT 56DISWAIT4TXMARGIN 57DISWAIT 58DIS2DETECT 59HOTRESETTS1 5AHOTRESETDONE 5BHOTRESETEIDLE 5CRECOVERYRCVRWAIT 5DRECOVERYRCVRMARGINCNT 5ERECOVERYRCVRLOCK 5FRECOVERYDESKEW 60RECOVERYRCVRCFG 61RECOVERYSPEED 62RECOVERYSPEEDTXEIDLE 63RECOVERYSPEEDRXEIDLE 64RECOVERYSPEEDREADY 65RECOVERYIDLE 66RECOVERYEXITDETECT 67RECOVERYLNK2DETECT 68RECOVERYEXITLPBK 69RECOVERYEXITL0 6ARECOVERYEXITDIS 6BRECOVERYEXITRST Note: This register field could be used by REUT software to monitor the link LTSSM substates.
23	0h RO	**Reserved (RSVD):** Reserved
22:19	0h RO/V	**Link Status (LNKSTAT):** During Link initialization the Link will always traverse this list of state from the top (0000) to the bottom of the list (0111). One or more power management states may be skipped, but the direction of list traversal will remain the same. 0000 Link Down 0001 : Link Retrain 0011 : L1 0100 : L2 0101 : L3 0111 : L0 (Link Up) 1000 : L0s (Transmit [amp] Receive) 1001 : L0s (Transmit only) 1010 : L0s (Receive only) All others reserved
18:17	0h RO/V	**Replay Number (REPLAYNUM):** Number of times the Retry Buffer has been replayed since the last Link initialization / re-training. When the Data Link Layer has replayed the contents of the Retry Buffer four times a Link re-training will be initiated which will reset this value back to zero.
16	0h RO/V	**Data Link Layer Retry (DLLRETRY):** Indicates when the Data Link Layer has received a corrupted TLP or has detected a dropped packet and is currently waiting for the remote agent to re-transmit the corrupted/dropped packet. The value of Next Receive Sequence Number will be the sequence number associated with the corrupted packet.
15:12	0h RO/V	**Lane Status (LANESTAT):** Indicates which lanes are trained. A '1' indicates that the corresponding lane is trained (i.e. bit 0 = '1' means lane 0 is trained).
11:0	0h RO/V	**Next Transmitted Sequence Number (NXTTXSEQNUM):** This is the sequence number to be applied to and pre-pended to the next outgoing TLP.

18.66 PCI Express Status 2 (PCIESTS2) — Offset 32Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 32Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**PCIe Port 3 Non-Common Clock With SSC Mode Enable Strap (P3PNCCWSSCMES):** '0': PCIe port 3 is not enabled to operate in non-common clock mode with SSC enabled. '1': PCIe port 3 is enabled to operate in non-common clock mode with SSC enabled.
30	0h RO/V	**PCIe Port 2 Non-Common Clock With SSC Mode Enable Strap (P2PNCCWSSCMES):** '0': PCIe port 2 is not enabled to operate in non-common clock mode with SSC enabled. '1': PCIe port 2 is enabled to operate in non-common clock mode with SSC enabled.
29	0h RO/V	**PCIe Port 1 Non-Common Clock With SSC Mode Enable Strap (P1PNCCWSSCMES):** '0': PCIe port 1 is not enabled to operate in non-common clock mode with SSC enabled. '1': PCIe port 1 is enabled to operate in non-common clock mode with SSC enabled.
28	0h RO/V	**PCIe Port 0 Non-Common Clock With SSC Mode Enable Strap (P0PNCCWSSCMES):** '0': PCIe port 0 is not enabled to operate in non-common clock mode with SSC enabled. '1': PCIe port 0 is enabled to operate in non-common clock mode with SSC enabled.
27:16	0h RO/V	**Next Receive Sequence Number (NXTRCVSEQ):** This is the sequence number associated with the TLP that is expected to be received next.
15:12	0h RW/1C/V	**Cause of Last Recovery Event (CLRE):** Cause of Last Recovery Event (CLRE): This field logs the cause of the entry to Recovery from L0. Only the first cause of Recovery is captured, until the register is cleared. Encoding Recovery Event 0000 No Recovery. 0001 Recovery entry triggered by remote device. 0010 Link Layer initiated Link Retrain due to error. 0011 De-skew buffer full. 0100 L0s exit time-out. 0101 Elastic Buffer overrun/underrun. 0110 Triggered by speed change. 0111 Link upconfiguration/downconfiguration. 1000 L0 Electrical Idle Inference. 1001 Any of the Link Retrain, CMM Start, Hot Reset, Link Disable, REUT Loopback Master or REUT Forced Loopback Master bit set. 1010 Received EIOS for RXL0s entry when ASPM L0s is disabled. 1011 Entry to Recovery from RXL0s due to PME timeout. Others Reserved.
11:0	0h RO/V	**Last Acknowledged Sequence Number (LASTACKSEQNUM):** This is the sequence number associated with the last acknowledged TLP.

18.67 PCI Express Compliance Measurement Mode (CMM) Port Control (PCIECMMPC) — Offset 330h

Note that selecting a lane number that does not exist for a port may result in undefined behavior.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 330h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GEN3 Intel CMM Scrambler Bypass (G3ICMMSB):** GEN3 Intel CMM Scrambler Bypass(G3ICMMSB): When set, the Intel CMM pattern will bypass scrambling in GEN3. This bit does not impact non Intel CMM pattern. The TSx and SOS prior to Intel CMM will still be scrambled normally. Note: This bit must be set prior to enabling Intel CMM, by setting the PCIECMMPC.START. Note: When operating in Mobile Express mode, this field is not applicable.
30	0h RO	**Reserved (RSVD):** Reserved
29	0h RW	**CMM Symbol[3] Select (SYM3SEL):** 0: selects CMM Symbol [lb]3[rb] to a control character 1: selects CMM Symbol [lb]3[rb] as a data character

Bit Range	Default & Access	Field Name (ID): Description
28	0h RW	**CMM Symbol[2] Select (SYM2SEL):** 0: selects CMM Symbol [lb]2[rb] to a control character 1: selects CMM Symbol [lb]2[rb] as a data character
27	0h RW	**CMM Symbol[1] Select (SYM1SEL):** 0: selects CMM Symbol [lb]1[rb] to a control character 1: selects CMM Symbol [lb]1[rb] as a data character
26	0h RW	**CMM Symbol[0] Select (SYM0SEL):** 0: selects CMM Symbol [lb]0[rb] to a control character 1: selects CMM Symbol [lb]0[rb] as a data character
25:24	0h RW	**CMM Sync Header (CMMSH):** CMM Sync Header(CMMSH): Specifies the Sync Header for the Intel CMM pattern specified in PCIECMMSB. Note: Due to implementation limitation, only a value of 10b is supported. All the other values are not supported.
23:22	0h RO/V	**CMM Error Lane Number (ERRLANENUM):** This field contains the lane number of the failing lane. Only valid when CMM Error Detected is 1.
21:16	0h RO	**Reserved (RSVD_1):** Reserved
15:13	0h RO/V	**CMM Invert (INVERT):** Indicates which lanes are inverted 000: No inversion 001: Lanes 0 010: Lanes 1 011: Lanes 2 100: Lanes 3 This field is only valid when CMM Error Detected (bit 7) is asserted. Additionally, when CMM Error Detected is asserted this field is locked (will not be updated)
12:10	0h RO/V	**CMM Symbol Error Number Invert (SYMERRNUMINV):** Indicates which register number miscompared on the failing lane, if the failing lane was an inverted lane. Only valid when CMM Error Detected is 1. 000: CMM Data D0 001: CMM Data D0 010: CMM Data D0 011: CMM Data D1 100: CMM Data D2 101: CMM Data D3 110: CMM Data D0 111: CMM Data D0
9:8	0h RO/V	**CMM Symbol Error Number (SYMERRNUM):** Indicates which register number miscompared on the failing lane, if the failing lane was not inverted. Only valid when CMM Error Detected is 1. 00: CMM Data 0 01: CMM Data 1 10: CMM Data 2 11: CMM Data 3
7	0h RW/1C/V	**CMM Error Detected (ERRDET):** 1: An error was detected 0: No error detected Note: This bit will be shadowed to an observability pin that can be used for IRQ generation.
6:5	0h RW	**Select Lane Number to be inverted for CMM (SLNINVCMM):** Select Lane Number to be inverted for CMM
4	0h RW	**CMM AutoInvert (AUTOINVERT):** 1: CMM autosequences through the inversion 0: CMM does not sequence inversion
3	0h RO/V	**CMM Status (STAT):** This bit is set when the CMM Start bit is set and cleared when the CMM mode has been entered successfully. 0: Compliance Measurement Mode is not active or CMM mode has been entered successfully. 1: Set as a result of CMM Start bit being set.
2	0h RW	**CMM Invert Enable (INVEN):** 1: Enables the Inversion of the lane 0: Lane not inverted
1	0h RW	**Reserved (RSVD_2):** Reserved
0	0h RW/L	**CMM Start (START):** 1: Start CMM 0: Stop CMM

18.68 PCI Express Compliance Measurement Mode Symbol Buffer (PCIECMMSB) — Offset 334h

Size:32 bits

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 334h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**CMM Data [3] (DATA3):** This character contains CMM Data [lb]3[rb] that will be transmitted on the link.
23:16	0h RW	**CMM Data [2] (DATA2):** This character contains CMM Data [lb]2[rb] that will be transmitted on the link.
15:8	0h RW	**CMM Data [1] (DATA1):** This character contains CMM Data [lb]1[rb] that will be transmitted on the link.
7:0	0h RW	**CMM Data [0] (DATA0):** This character contains CMM Data [lb]0[rb] that will be transmitted on the link.

18.69 PTM Propagation Delay (PTMPD) — Offset 390h

Size:32 bits

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 390h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Current PTM Propagation Delay Value (CPTMPDV):** Current PTM Propagation Delay Value(CPTMPDV): This field reports the current PTM Propagation Delay value captured from the last successful PTM dialog.

18.70 PTM Lower Local Master Time (PTMLLMT) — Offset 394h

Size:32 bits

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 394h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Current PTM Lower Local Master Time Value (CPTMLLMTV):** Current PTM Lower Local Master Time Value(CPTMLLMTV): This field reports the lower fields bits 31:0 of the Local TSC time value.

18.71 PTM Upper Local Master Time (PTMULMT) — Offset 398h

Size:32 bits

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 398h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Current PTM Upper Local Master Time Value (CPTMULMTV):** Current PTM Upper Local Master Time Value(CPTMULMTV): This field reports the upper fields bits 63:32 of the Local TSC time value.

18.72 PTM Pipe Stage Delay Configuration 1 (PTMPSDC1) — Offset 39Ch

Size:32 bits

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 39Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**GEN1 X2 RX Pipe Stage Delay (G1X2RPSD):** GEN1 X2 RX Pipe Stage Delay(G1X2RPSD): 00h: 1 link clock delay. 01h: 2 link clock delay. 02h: 3 link clock delay. : : FFh: 256 link clock delay.
23:16	0h RW	**GEN1 X2 TX Pipe Stage Delay (G1X2TPSD):** GEN1 X2 TX Pipe Stage Delay(G1X2TPSD): 00h: 1 link clock delay. 01h: 2 link clock delay. 02h: 3 link clock delay. : : FFh: 256 link clock delay.
15:8	0h RW	**GEN1 X1 RX Pipe Stage Delay (G1X1RPSD):** GEN1 X1 RX Pipe Stage Delay(G1X1RPSD): 00h: 1 link clock delay. 01h: 2 link clock delay. 02h: 3 link clock delay. : : FFh: 256 link clock delay.
7:0	0h RW	**GEN1 X1 TX Pipe Stage Delay (G1X1TPSD):** GEN1 X1 TX Pipe Stage Delay(G1X1TPSD): 00h: 1 link clock delay. 01h: 2 link clock delay. 02h: 3 link clock delay. : : FFh: 256 link clock delay.

18.73 PTM Pipe Stage Delay Configuration 2 (PTMPSDC2) — Offset 3A0h

Size:32 bits

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 3A0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**GEN2 X1 RX Pipe Stage Delay (G2X1RPSD):** GEN2 X1 RX Pipe Stage Delay(G2X1RPSD): 00h: 1 link clock delay. 01h: 2 link clock delay. 02h: 3 link clock delay. : : FFh: 256 link clock delay.

Bit Range	Default & Access	Field Name (ID): Description
23:16	0h RW	**GEN2 X1 TX Pipe Stage Delay (G2X1TPSD):** GEN2 X1 TX Pipe Stage Delay(G2X1TPSD): 00h: 1 link clock delay. 01h: 2 link clock delay. 02h: 3 link clock delay. : : FFh: 256 link clock delay.
15:8	0h RW	**GEN1 X4 RX Pipe Stage Delay (G1X4RPSD):** GEN1 X4 RX Pipe Stage Delay(G1X4RPSD): 00h: 1 link clock delay. 01h: 2 link clock delay. 02h: 3 link clock delay. : : FFh: 256 link clock delay.
7:0	0h RW	**GEN1 X4 TX Pipe Stage Delay (G1X4TPSD):** GEN1 X4 TX Pipe Stage Delay(G1X4TPSD): 00h: 1 link clock delay. 01h: 2 link clock delay. 02h: 3 link clock delay. : : FFh: 256 link clock delay.

18.74 PTM Pipe Stage Delay Configuration 3 (PTMPSDC3) — Offset 3A4h

Size:32 bits

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 3A4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**GEN2 X4 RX Pipe Stage Delay (G2X4RPSD):** GEN2 X4 RX Pipe Stage Delay(G2X4RPSD): 00h: 1 link clock delay. 01h: 2 link clock delay. 02h: 3 link clock delay. : : FFh: 256 link clock delay.
23:16	0h RW	**GEN2 X4 TX Pipe Stage Delay (G2X4TPSD):** GEN2 X4 TX Pipe Stage Delay(G2X4TPSD): 00h: 1 link clock delay. 01h: 2 link clock delay. 02h: 3 link clock delay. : : FFh: 256 link clock delay.
15:8	0h RW	**GEN2 X2 RX Pipe Stage Delay (G2X2RPSD):** GEN2 X2 RX Pipe Stage Delay(G2X2RPSD): 00h: 1 link clock delay. 01h: 2 link clock delay. 02h: 3 link clock delay. : : FFh: 256 link clock delay.
7:0	0h RW	**GEN2 X2 TX Pipe Stage Delay (G2X2TPSD):** GEN2 X2 TX Pipe Stage Delay(G2X2TPSD): 00h: 1 link clock delay. 01h: 2 link clock delay. 02h: 3 link clock delay. : : FFh: 256 link clock delay.

18.75 PTM Pipe Stage Delay Configuration 4 (PTMPSDC4) — Offset 3A8h

Size:32 bits

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 3A8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**GEN3 X2 RX Pipe Stage Delay (G3X2RPSD):** GEN3 X2 RX Pipe Stage Delay(G3X2RPSD): 00h: 1 link clock delay. 01h: 2 link clock delay. 02h: 3 link clock delay. : : FFh: 256 link clock delay.

Bit Range	Default & Access	Field Name (ID): Description
23:16	0h RW	**GEN3 X2 TX Pipe Stage Delay (G3X2TPSD):** GEN3 X2 TX Pipe Stage Delay(G3X2TPSD): 00h: 1 link clock delay. 01h: 2 link clock delay. 02h: 3 link clock delay. : : FFh: 256 link clock delay.
15:8	0h RW	**GEN3 X1 RX Pipe Stage Delay (G3X1RPSD):** GEN3 X1 RX Pipe Stage Delay(G3X1RPSD): 00h: 1 link clock delay. 01h: 2 link clock delay. 02h: 3 link clock delay. : : FFh: 256 link clock delay.
7:0	0h RW	**GEN3 X1 TX Pipe Stage Delay (G3X1TPSD):** GEN3 X1 TX Pipe Stage Delay(G3X1TPSD): 00h: 1 link clock delay. 01h: 2 link clock delay. 02h: 3 link clock delay. : : FFh: 256 link clock delay.

18.76 PTM Pipe Stage Delay Configuration 5 (PTMPSDC5) — Offset 3ACh

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 3ACh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:8	0h RW	**GEN3 X4 RX Pipe Stage Delay (G3X4RPSD):** GEN3 X4 RX Pipe Stage Delay(G3X4RPSD): 00h: 1 link clock delay. 01h: 2 link clock delay. 02h: 3 link clock delay. : : FFh: 256 link clock delay.
7:0	0h RW	**GEN3 X4 TX Pipe Stage Delay (G3X4TPSD):** GEN3 X4 TX Pipe Stage Delay(G3X4TPSD): 00h: 1 link clock delay. 01h: 2 link clock delay. 02h: 3 link clock delay. : : FFh: 256 link clock delay.

18.77 PTM Extended Config (PTMECFG) — Offset 3B0h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 3B0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:21	0h RO	**Reserved (RSVD):** Reserved
20:18	0h RW	**Periodic Local TSC Link Fetch Frequency (PLTLFF):** Periodic Local TSC Link Fetch Frequency (PLTLFF): When this register is programmed to a non-zero values, the Local TSC Link Clock would perform a periodic fetch to obtain the latest TSC from the Local TSC XTAL Clock domain. This mechanism would ensure the Root Port Local TSC Link is always synchronized with the actual TSC as Link Clock domain is able to drift due to SSC. 000: Disable this feature. 001: Always pull without waiting for expiration. 010: Every 8 clocks 011: Every 16 clocks 100: Every 32 clocks 101: Every 64 clocks 110: Every 128 clocks 111: Every 256 clocks This register is only available in Port 1. Note: Software is expected to program this register prior to setting PTM Enable.

Bit Range	Default & Access	Field Name (ID): Description
17:15	0h RW/1C/V	**Global Time Fetch Retry Counter (GTFRC):** Global Time Fetch Retry Counter. This register is incremented when the Root Port detected a retry on each Global Time Fetch on IOSF Sideband. The Root Port would increment the value of this register whenever ARU re-sends a LocalSync message. If more than 7 Retries are detected during the Global Time Fetch, Root Port would keep the value of this register to 111 (max) value. Software is expected to write 111 to this register to clear the entire field to 0. Note: For each x4 instance, only the value from Port 1 is used.
14:13	0h RW/1C/V	**Global Time Fetch Fail Counter (GTFFC):** Global Time Fetch Fail Counter. This register is incremented when the Root Port detected a fail on each Global Time Fetch on IOSF Sideband. The Root Port would increment the value of this register whenever ARU sends a SyncComp with the Fail status. If more than 3 failures are detected in the Global Time Fetch, Root Port would keep the value of this register to 111 (max) value. Software is expected to write 11 to this register to clear the entire field to 0. Note: For each x4 instance, only the value from Port 1 is used.
12	0h RO/V	**Global Time Fetch Status Pending Completion (GTFSPC):** Global Time Fetch Status Pending Completion. This register is set to 1 by the Root Port when it is in progress of fetching the Global Time from ARU. Note: For each x4 instance, only the value from Port 1 is used.
11:9	0h RW	**Periodic Global Time Stamp Counter Fetch Frequency (PGTSCFF):** Periodic Global Time Stamp Counter Fetch Frequency (PGTSCFF) : This field determine the frequency the Root Port would autonomously fetch the Global Time Stamp Counter. 00: 10us 01: 100us 10: 500us 10: 1ms Software is expected to program this bit first before programming the PGTSCFE register. Attribute: Dynamic Note: For each x4 instance, only the value from Port 1 is used.
8	0h RW	**Periodic Global Time Stamp Counter Fetch Enable (PGTSCFE):** Periodic Global Time Stamp Counter Fetch Enable (PGTSCFE) : When this bit set, the Controller will re-fetch the Global Time from the Always Running Unit (ARU). Once Fetch is completed, the Controller would update all the Local TSC with the newly fetch Global Time. If any PTM dialog is initiated while the re-fetch occurred, the Controller would use the existing Local TSC timers. Hardware would clear this bit upon completed fetching the Global Time. Attribute : Dynamic Note: For each x4 instance, only the value from Port 1 is used.
7	0h RW	**Trigger Global Time Stamp Counter Fetch Enable (TGTSCFE):** Trigger Global Time Stamp Counter Fetch Enable (TGTSCFE) : When this bit set, the Controller will re-fetch the Global Time from the Always Running Unit (ARU). Once Fetch is completed, the Controller would update all the Local TSC with the newly fetch Global Time. If any PTM dialog is initiated while the re-fetch occurred, the Controller would use the existing Local TSC timers. Hardware would clear this bit upon completed fetching the Global Time. Software can only set this register if PGTSCFE is not set. Attribute : Dynamic Note: For each x4 instance, only the value from Port 1 is used.
6	0h RW	**PTM Request Periodic ACK Enable (PTMRPAE):** PTM Request Periodic ACK Enable (PTMRPAE) : When this register is set to 1, whenever a valid PTM request TLP is received, the Link Layer would transmit multiple ACK DLLPs corresponding to the PTM Request message. The number of ACK DLLP that the Link Layer would transmit is based on the PTMRNOPAD register. Attribute : Static Note: For each x4 instance, only the value from Port 1 is used.
5:4	0h RW	**PTM Request Number Of Periodic ACK DLLP (PTMRNOPAD):** PTM Request Number Of Periodic ACK DLLP (PTMRNOPAD) : When PTMRPAE is enable, whenever a valid PTM Request message is received, the Link Layer would transmit multiple ACK DLLP corresponding to the receiving of the PTM Request message. This register define the number of DLLP ACK will be transmitted as high priority. 00 - TX 1 DLLP ACK 01 - TX 2 DLLP ACK 10 - TX 3 DLLP ACK 11 - TX 4 DLLP ACK Attribute : Static Note: For each x4 instance, only the value from Port 1 is used.
3:0	0h RW	**IOSF Max Allowed Delay programming (IOSFMADP):** IOSF Max Allowed Delay programming (IOSFMADP): bits Status 0000 Bound Range Low 0001 Bound Range 2 1000 Bound Range Max others reserved

18.78 PTM Lower T2 Time Stamp (PTMLT2TSTMP) — Offset 3B4h

Size:32 bits

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 3B4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Latest Captured Lower T2 TimeStamp (LCLT2TS):** Latest Captured Lower T2 TimeStamp (LCLT2TS). This field shows the latest lower 32-bit of T2 TimeStamp captured by the Root Port in TSC Clock Domain when the Root Port received a valid PTM Request message. The renewable T2 TimeStamp due to a duplicate PTM Request would also be reflected in this field.

18.79 PTM Upper T2 Time Stamp (PTMUT2TSTMP) — Offset 3B8h

Size:32 bits

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 3B8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Latest Captured Upper T2 TimeStamp (LCUT2TS):** Latest Captured Upper T2 TimeStamp (LCUT2TS). This field shows the latest upper 32-bit of T2 TimeStamp captured by the Root Port in TSC Clock Domain when the Root Port received a valid PTM Request message. The renewable T2 TimeStamp due to a duplicate PTM Request would also be reflected in this field.

18.80 Strap and Fuse Configuration 2 (STRPFUSECFG2) — Offset 414h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 414h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**mod-PHY Power Gating Disable Fuse (MPHYPGD):** 0: mod-PHY power gating is enabled. 1: mod-PHY power gating is disabled. Note: Prior to fuse pull, the default of this bit is specified in the 'Reset' column of this field. The default value will reflect the fuse value once fuse pull is done.
30:0	0h RO	**Reserved (RSVD):** Reserved

18.81 Thermal and Power Throttling (TNPT) — Offset 418h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 418h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**Throttle Period (TP):** Throttle Period (TP): If any of the TNPT.DRXLTE or TNPT.DTXLTE bit is '1, this field defines the duration in milliseconds that defines the Throttling Window. When TNPT.TTG is set to 0, the effective Throttling Period is: 00h: 1 ms 01h: 2 ms : : FFh: 256 ms Note: The Throttle Period will have an uncertainty of +/-1 ms. When TNPT.TTG is set to 1, the effective Throttling Period is: 00h: 100 us 01h: 200 us : : FFh: 25.6 ms Note: The Throttle Period will have an uncertainty of +/-100 us. Note: To change the Link Throttling control bits, the DRXLTE and DTXLTE bits must be disabled first, change the value, and then re-enable the Link throttling. Note: If TNPT.TT is programmed to a value bigger than TNPT.TP, the hardware behavior is undefined.
23:16	0h RW	**Throttle Time (TT):** Throttle Time (TT): If any of the TNPT.DRXLTE or TNPT.DTXLTE bit is '1, this field defines the period of the Throttling Zone within the Throttling Window specified by TNPT.TP. The value specified in this field will be multiplied by the respective multiplier in TNPT.TSLxM fields depending on the throttling severity indication received together with the Throttling State change indication. When TNPT.TTG is set to 0, the effective Throttle Time is: 00h: 1 ms 01h: 2 ms : 3Fh: 64 ms Others: Alias to 3Fh. Note: The Throttle Period will have an uncertainty of +/-1 ms. When TNPT.TTG is set to 1, the effective Throttle Time is: 00h: 100 us 01h: 200 us : 3Fh: 6.4 ms Note: The Throttle Period will have an uncertainty of +/-100 us. Note: If the reserved encoding is programmed to this field, hardware will behave the same as if the field is programmed to 3Fh. Note: Since the design is using a 1 ms tick for this timer, the Throttle Time will have an uncertainty of +/-1 ms. Note: To change the Link Throttling control bits, the DRXLTE and DTXLTE bits must be disabled first, change the value, and then re-enable the Link throttling. Note: If TNPT.TT is programmed to a value bigger than TNPT.TP, the hardware behavior is undefined.
15:12	0h RO	**Reserved (RSVD):** Reserved
11:10	0h RW	**Throttling Severity Level 3 Multiplier (TSL3M):** Throttling Severity Level 3 Multiplier (TSL3M): This register determines the multiplier to be used together with the TNPT.TT field to specify the period of the Throttling Zone within the Throttling Window. 00b: x1 01b: x2 10b: x4 11b: Always throttling. Note: To change the Link Throttling control bits, the DRXLTE and DTXLTE bits must be disabled first, change the value, and then re-enable the Link throttling.
9:8	0h RW	**Throttling Severity Level 2 Multiplier (TSL2M):** Throttling Severity Level 2 Multiplier (TSL2M): This register determines the multiplier to be used together with the TNPT.TT field to specify the period of the Throttling Zone within the Throttling Window. 00b: x1 01b: x2 10b: x4 11b: Always throttling. Note: To change the Link Throttling control bits, the DRXLTE and DTXLTE bits must be disabled first, change the value, and then re-enable the Link throttling.
7:6	0h RW	**Throttling Severity Level 1 Multiplier (TSL1M):** Throttling Severity Level 1 Multiplier (TSL1M): This register determines the multiplier to be used together with the TNPT.TT field to specify the period of the Throttling Zone within the Throttling Window. 00b: x1 01b: x2 10b: x4 11b: No throttling. Note: To change the Link Throttling control bits, the DRXLTE and DTXLTE bits must be disabled first, change the value, and then re-enable the link throttling
5:4	0h RW	**Throttling Severity Level 0 Multiplier (TSL0M):** Throttling Severity Level 0 Multiplier (TSL0M): This register determines the multiplier to be used together with the TNPT.TT field to specify the period of the Throttling Zone within the Throttling Window. 00b: x1 01b: x2 10b: x4 11b: No throttling. Note: To change the Link Throttling control bits, the DRXLTE and DTXLTE bits must be disabled first, change the value, and then re-enable the Link throttling.
3	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Throttling Timer Granuality (TTG):** Throttling Timer Granuality (TTG): This register determines the granuality of the Thermal Throttling timers. This provides a smaller granuality
1	0h RW	**Dynamic RX Link Throttling Enable (DRXLTE):** Dynamic RX Link Throttling Enable (DRXLTE): '0b: Dynamic Link RX Throttling mechanism is disabled. '1b: Dynamic Link RX Throttling mechanism is enabled. PCIe Root Port will induce the link to enter RXL0s. The duty cycle of the throttling window is configurable based on the throttling severity. Note: This field can only be set if the remote component supports TXL0s.
0	0h RW	**Dynamic TX Link Throttling Enable (DTXLTE):** Dynamic TX Link Throttling Enable (DTXLTE): '0b: Dynamic Link TX Throttling mechanism is disabled. '1b: Dynamic Link TX Throttling mechanism is enabled. PCIe Root Port will induce the link to enter TXL0s. The duty cycle of the throttling window is configurable based on the throttling severity. Note: This field can only be set if the remote component supports TXL0s.

18.82 Dynamic Lane Switch (DYNLNSW) — Offset 41Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 41Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	**Reserved (RSVD):** Reserved
0	0h RW	**Hardware Re-Do Preset to Coefficient Mapping Query After Lane Switching (HWRP2CM):** Hardware Re-Do Preset to Coefficient Mapping Query After Lane Switching (HWRP2CM): When this bit is set, the PCIe-SIP Controller would query the Preset to Coefficient mapping through the PIPE GetLocalPresetCoefficients and LocalTxCoefficientsValid interface whenever the Lane Switch ownership has transitioned to PCIe (from another Controller). Note that if this bit is set while the HPCMQE bit is set, the PCIe-SIP Controller would only perform the query once. Unlike the HPCMQE bit, the PCIe-SIP Controller would not clear this bit after completing the query over the PIPE interface. Register Attribute: Static.

18.83 Power Control Enable (PCE) — Offset 428h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 428h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RO	**Reserved (RSVD):** Reserved
5	0h RW	**Hardware Autonomous Enable (HAE):** Hardware Autonomous Enable (HAE): If set, and the corresponding per-LTSSM state power gating enable bit is also set, then controller power gating will be done when the controller is idle and the controller power gating condition is met in that particular LTSSM state. Refer to PCIEPMECTL2 register for the per-LTSSM state power gating enable bit. If either this bit. is not set or the corresponding per-LTSSM state power gating enable bit is not set, then controller power gating will not be done in that LTSSM state. Note: For each x4 instance, only the value from Port 0 is used. NOTE: If this bit is set, then bits[lb]2:0[rb] must be '000.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RO	**Reserved (RSVD_1):** Reserved
3	0h RW/L	**Sleep Enable (SE):** Sleep Enable (SE): If clear, Sleep indication to the retention flops will never assert. If set, Sleep indication will be assert to the retention flops as part of the hardware autonomous controller power gating entry flow.
2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO	**Reserved (RSVD_3):** Reserved
0	0h RW	**PMC Request Enable (PMCRE):** PMC Request Enable (PMCRE): When set, the controller will only power gate when pmc_[lt]ip[gt]_sw_pg_req_b = '0 and hardware autonomous controller power gating conditions are met. When clear, controller will power gate immediately when the hardware autonomous controller power gating conditions are met regardless of the state of pmc_[lt]ip[gt]_sw_pg_req_b.

18.84 PGCB Control1 (PGCBCTL1) — Offset 42Ch

This register specifies the minimum number of delay clocks the PGCB should wait between various states. Refer to the PGCB for the description of the individual fields. Note: For each x4 instance, only the value from Port 0 is used.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 42Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:28	0h RW	**cfg_trstup2frcclks (TRSTUP2FRCCLKS):** cfg_trstup2frcclks(cfg_trstup2frcclks): '00: 1 clock '01: 2 clocks '10: 8 clocks '11: 256 clocks
27:26	0h RW	**cfg_tclksonack_cp (TCLKSONACK_CP):** cfg_tclksonack_cp(cfg_tclksonack_cp): '00: 1 clock '01: 2 clocks '10: 8 clocks '11: 256 clocks
25:24	0h RO	**Reserved (RSVD_1):** Reserved
23:22	0h RO	**Reserved (RSVD_2):** Reserved
21:20	0h RW	**cfg_tpokup (TPOKUP):** cfg_tpokup(cfg_tpokup): '00: 1 clock '01: 2 clocks '10: 8 clocks '11: 256 clocks
19:18	0h RW	**cfg_tpokdown (TPOKDOWN):** cfg_tpokdown(cfg_tpokdown): '00: 1 clock '01: 2 clocks '10: 8 clocks '11: 256 clocks
17:16	0h RW	**cfg_tlatchdis (TLATCHDIS):** cfg_tlatchdis(cfg_tlatchdis): '00: 1 clock '01: 2 clocks '10: 8 clocks '11: 256 clocks
15:14	0h RW	**cfg_tsleepinactiv (TSLEEPINACTIV):** cfg_tsleepinactiv(cfg_tsleepinactiv): '00: 1 clock '01: 2 clocks '10: 8 clocks '11: 256 clocks
13:12	0h RW	**cfg_tinaccrstup (TINACCRSTUP):** cfg_tinaccrstup(cfg_tinaccrstup): '00: 1 clock '01: 2 clocks '10: 8 clocks '11: 256 clocks

Bit Range	Default & Access	Field Name (ID): Description
11:10	0h RW	**cfg_taccrstup (TACCRSTUP):** cfg_taccrstup(cfg_taccrstup): '00: 1 clock '01: 2 clocks '10: 8 clocks '11: 256 clocks
9:8	0h RW	**cfg_tlatchen (TLATCHEN):** cfg_tlatchen(cfg_tlatchen): '00: 1 clock '01: 2 clocks '10: 8 clocks '11: 256 clocks
7:6	0h RW	**cfg_tdeisolate (TDEISOLATE):** cfg_tdeisolate(cfg_tdeisolate): '00: 1 clock '01: 2 clocks '10: 8 clocks '11: 256 clocks
5:4	0h RW	**cfg_trstdown (TRSTDOWN):** cfg_trstdown(cfg_trstdown): '00: 1 clock '01: 2 clocks '10: 8 clocks '11: 256 clocks
3:2	0h RW	**cfg_tisolate (TISOLATE):** cfg_tisolate(cfg_tisolate): '00: 1 clock '01: 2 clocks '10: 8 clocks '11: 256 clocks
1:0	0h RW	**cfg_tsleepact (TSLEEPACT):** cfg_tsleepact(cfg_tsleepact): '00: 1 clock '01: 2 clocks '10: 8 clocks '11: 256 clocks

18.85 PGCB Control2 (PGCBCTL2) — Offset 430h

This register specifies the minimum number of delay clocks the PGCB should wait between various states. Note: For each x4 instance, only the value from Port 0 is used.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 430h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD):** Reserved
7:6	0h RW	**cfg_trsvd4 (TRSVD4):** cfg_trsvd4(cfg_trsvd4): '00: 1 clock '01: 2 clocks '10: 8 clocks '11: 256 clocks
5:4	0h RW	**cfg_trsvd3 (TRSVD3):** cfg_trsvd3(cfg_trsvd3): '00: 1 clock '01: 2 clocks '10: 8 clocks '11: 256 clocks
3:2	0h RW	**cfg_trsvd2 (TRSVD2):** cfg_trsvd2(cfg_trsvd2): '00: 1 clock '01: 2 clocks '10: 8 clocks '11: 256 clocks
1:0	0h RO	**Reserved (RSVD_1):** Reserved

18.86 Equalization Configuration 1 (EQCFG1) — Offset 450h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 450h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO/V	**REC:** Recovery Entry Count (REC): This field indicates the value of the Recovery Entry Counter. This is a 1-based counter. Software must read this register multiple times. The value is valid only if the same value is read out on both of the reads.
23	0h RW	**Recovery Entry and Idle Framing Error Count Enable (REIFECE):** Recovery Entry and Idle Framing Error Count Enable (REIFECE): This bit, when set by software turns on the Recovery Entry Counter and the per-lane Idle Framing Error Counter. The counters are reset when this bit is cleared. This bit is expected to be used by the Software Preset/Coefficient Search tool but is not precluded to be used for other debug purpose. The value of the Recovery Entry Count can be read through EQCFG1.REC field. The value of the Idle Framing Error Count can be read through the Monitor Mux register.
22	0h RW	**Quiesce Guarantee (QG):** Quiesce Guarantee (QG): When set, the Quiesce Guarantee bit in the transmitted TS2 Ordered Set will be set in Recovery.RcvrCfg. When clear, the Quiesce Guarantee bit in the transmitted TS2 Ordered Set will be clear. In all other states, the Quiesce Guarantee bit is Reserved.
21	0h RW	**Link Equalization Request SMI Enable (LERSMIE):** Link Equalization Request SMI Enable (LERSMIE): When set, this bit enables the generation of an SMI to indicate that the Link Equalization Request bit has been set. This mode is meant for survivability purpose such that BIOS can be invoked to address the Re-Equalization request.
20	0h RW	**Reset EIEOS Interval Count (REIC):** Reset EIEOS Interval Count(REIC): When set, allows the root port to restrict the device from sending EIEOS until after 65536 TS1 Ordered Sets have been transmitted in Phase 3 of the Link Equalization, in the window when the receiver is evaluating the remote transmitter settings..
19	0h RW	**Link Equalization Bypass (LEB):** Link Equalization Bypass (LEB): When set, the root port will never initiate entry to Recovery.Equalization state. This includes never send EQ TS2 in Recovery.RcvrCfg that could cause the device to set start_equalization_w_preset variable. Note: This bit only affects the initial autonomous transition to Link Equalization state when equalization_done_8GT_data_rate = 0. This bit does not affect the software-direction to re-perform Link Equalization.
18	0h RW	**Link Equalization Phase 2 and 3 Bypass (LEP23B):** Link Equalization Phase 2 and 3 Bypass(LEP23B): When set, bypasses the Phase 2 and Phase 3 of Link Equalization. Once Phase 1 is completed, Root Port transitions from Phase 1 directly to Recovery.RcvrLock.
17	0h RW	**Link Equalization 3 Bypass (LEP3B):** Link Equalization 3 Bypass(LEP3B): When set, bypasses the Phase 3 of Link Equalization. Once Phase 2 is completed, Root Port transitions from Phase 2 directly to Recovery.RcvrLock.
16	0h RW	**Remote Transmit Link Equalization Preset/Coefficient Evaluation Bypass (RTLEPCEB):** Remote Transmit Link Equalization Preset/Coefficient Evaluation Bypass (RTLEPCEB): When set, this bit disables the Hardware Autonomous Preset/Coefficient Search mechanism to search for the best Preset or Coefficient by traversing the Preset or Coefficient List and checking the receiver eye width margin for each of the settings. Instead, the Preset/Coefficient values used by the remote Transmitter will be accepted and the Link Equalization phase will be completed after one round of receiver link training, excluding margining. BIOS must ensure that the link runs in either GEN1 or GEN2 when the registers are configured. Once the register programming are done, BIOS should set the Perform Equalization bit to '1, Target Link Speed register to 0011b and initiate Link Retrain by setting the Retrain Link bit to 1b.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**Remote Transmitter Preset Coefficient Override Enable (RTPCOE):** Remote Transmitter Preset Coefficient Override Enable (RTPCOE): When set, this bit disables the hardware mechanism to search for the best Preset or Coefficient by traversing the Preset or Coefficient List and checking the receiver eye width margin for each of the settings. Instead, the Preset or Coefficient values specified by the override fields are used. If RTPCL1.PCM = 1, the Preset Override values for each lanes is derived from the following register fields: Lane 0: RTPCL1.RTPRECL0PL0. Lane 1: RTPCL1.RTPOSTCL0PL1. Lane 2: RTPCL1.RTPRECL1PL2. Lane 3: RTPCL1.RTPOSTCL1PL3. If RTPCL1.PCM = 0, the Coefficient Override values for each lanes is derived from the following register fields: Lane 0: RTPCL1.RTPRECL0PL0 and RTPCL1.RTPOSTCL0PL1. Lane 1: RTPCL1.RTPRECL1PL2 and RTPCL1.RTPOSTCL1PL3. Lane 2: RTPCL1.RTPRECL2PL4 and RTPCL2.RTPOSTCL2PL5. Lane 3: RTPCL2.RTPRECL3PL6 and RTPCL2.RTPOSTCL3PL7. BIOS must ensure that the corresponding RTPCL* registers above are programmed correctly prior to setting this bit. BIOS must ensure that the link runs in either GEN1 or GEN2 when the registers are configured. Once the register programming are done, BIOS should set the Perform Equalization bit to '1, Target Link Speed register to 0011b and initiate Link Retrain by setting the Retrain Link bit to 1b.
14	0h RW	**Link Equalization Request SCI Enable (LERSCIE):** Link Equalization Request SCI Enable (LERSCIE): When set, this bit enables the generation of an SCI to indicate that the Link Equalization Request bit has been set. This mode is meant for survivability purpose such that SCI handler can be invoked to address the Re-Equalization request.
13	0h RW/1S/V	**Hardware Preset to Coefficient Mapping Query Enable (HPCMQE):** Hardware Preset to Coefficient Mapping Query Enable(HPCMQE): When set, the controller will query the Preset to Coefficient mapping through the PIPE GetLocalPresetCoefficients and LocalTxCoefficientsValid interface whenever this bit transitions from 0 to 1. The default of this register bit is 1, indicating that the Preset to Coefficient mapping query will be done on the PIPE interface once coming out of reset. Controller will then update the Preset-Coefficient Mapping registers with the corresponding Coefficient, for each Preset. Controller will also update the LFFS Local LF and Local FS field with the local PHY LF and FS values. Hardware will clear this bit when the Preset to Coefficient mapping query over the PIPE interface is completed. If the Hardware Preset to Coefficient Mapping mechanism is never enabled, the value of the Preset to Coefficient mapping configured by BIOS through the Preset-Coefficient Mapping registers will be used instead of querying through the PIPE interface. Note: BIOS should check to ensure that this field is cleared before enabling Controller Power Gating or mod-PHY Power Gating.
12	0h RO/V	**Hardware Autonomous Equalization Done (HAED):** Hardware Autonomous Equalization Done(HAED): This bit will be cleared when Hardware Autonomous Preset/ Coefficient Search starts and will be set when Hardware Autonomous Preset/Coefficient Search is done. This bit is polled by software to ensure that the Hardware Autonomous Preset/Coefficient Search is done before proceeding with the next software sequencing. Some of the Hardware Autonomous Preset/ Coefficient search algorithm may involve the hardware initiating multiple speed change to allow multiple iterations of Link Equalization to be done with different Preset/Coefficient lists. This bit will remain cleared until the iterations are done.
11:8	0h RW	**Receiver Wait Time For New Equalization Value Evaluation (RWTNEVE):** Receiver Wait Time For New Equalization Value Evaluation (RWTNEVE): For Downstream Port: This field specifies the amount of time the receiver will wait after entering Phase 3 and sending the new Preset or Coefficient values through the TS1 Ordered Sets before validating the Block Alignment and eventually evaluate the incoming ordered sets (RXEqEval on the PIPE interface asserts). For Upstream Port: This field specifies the amount of time the receiver will wait after entering Phase 2 and sending the new Preset or Coefficient values through the TS1 Ordered Sets before validating the Block Alignment and eventually evaluate the incoming ordered sets (RXEqEval on the PIPE interface asserts). For both Upstream and Downstream Port, this field also specifies the amount of time the receiver will wait after entering Phase 1 before instructing the receiver to adapt to the incoming ordered sets. For Loopback Master: This field specifies the amount of time the receiver will wait after instructing the Loopback Slave to apply a specific Preset through EQ TS1. 0h: 500 ns. 1h: 1 us. 2h: 2 us. 3h: 3 us. 4h: 4 us. : : Fh:15 us.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**EQ TS2 in Recovery.ReceiverConfig Enable (EQTS2IRRC):** EQ TS2 in Recovery.ReceiverConfig Enable(EQTS2IRRC): When set, enables the transmitter to send EQ TS2 in Recovery.RcvrCfg state even when equalization_done_8GT_data_rate variable is 1b, provided that the Downstream Port advertised 8.0 GT/s data rate support in Recovery.RcvrLock, and 8.0 GT/s data rate support has been advertised in the Configuration.Complete or Recovery.RcvrCfg substates by the Upstream Port since exiting the Detect state, and eight consecutive TS1 or TS2 Ordered Sets were received on any configured Lane prior to entry to this substate with speed_change bit set to 1b. When clear, the transmitter can only send EQ TS2 if equalization_done_8GT_data_rate variable is 0b and the Downstream Port advertised 8.0 GT/s data rate support in Recovery.RcvrLock, and 8.0 GT/s data rate support has been advertised in the Configuration.Complete or Recovery.RcvrCfg substates by the Upstream Port since exiting the Detect state, and eight consecutive TS1 or TS2 Ordered Sets were received on any configured Lane prior to entry to this substate with speed_change bit set to 1b. When this bit is used, hardware must ensure that the start_equalization_w_preset variable are in the correct state to ensure that the components on both sides of the link are never out of sync.
6:4	0h RO	**Reserved (RSVD):** Reserved
3	0h RW	**Link EQ Phase 1 Transmit Coefficient Settling Policy (LEQP1TCSP):** Link EQ Phase 1 Transmit Coefficient Settling Policy(LEQP1TCSP): When operating in GEN3 data rate and there is a software/hardware request to re-perform Link Equalization through the Recovery.RcvrLock to Recovery.Equalization arc, PCIe spec requires that the downstream port transmitter switch to the setting specified by the Downstream Port Lane X Transmitter Preset registers in Phase 1. This switching is happening while the downstream port is still actively transmitting TS1 and the upstream port is only required to sample 2 TS1 to determine the next sub-state to transition to. Since the new coefficient setting can take up to 256 ns to settle, the 2 TS1 sampled by the upstream port may be incorrect causing the two LTSSM to be out of sync. When this bit is set, the RP will continue to send EIEOS until the local transmitter setting has settled (specified by PHYCTL2.TXCFGCHGWAIT) before sending TS1 as required in Recovery.Equalization Phase 1. When this bit is clear, the RP will send TS1 with EC = 01 in Recovery.Equalization Phase 1 even though the transmitter setting is still settling.
2	0h RW	**Multi-Fragment Linear and Nine-Tile List Enable (MFLNTL):** Multi-Fragment Linear and Nine-Tile List Enable(MFLNTL): When set in Hardware Autonomous Linear Preset/Coefficient Search mode, the full Preset/Coefficient List will be traversed in multiple fragments, where each fragments is done in separate entry to Recovery. This is used in the case where a longer dwelling time is required for a particular Preset/ Coefficient (configured through EQCFG2.PCET). Subsequent Preset/Coefficient entries within the list that could not be covered within that Recovery session will be covered in subsequent re-entries into Recovery. When set in Hardware Autonomous Nine-Tiles Search mode, the 9-tiles list that could not be covered within that Recovery session will be covered in subsequent re-entries into Recovery.
1	0h RW	**Transmitter Use Preset Policy (TUPP):** Transmitter Use Preset Policy(TUPP): This field applies to the Link Equalization Phase where the local transmitter setting is being adjusted. When set, the transmitted TS1 Use Preset bit will be set if the remote device requests the local transmitter to apply specific Preset(instead of Coefficient). When clear, the Use Preset bit will not be set in this case. Note: This bit must be set before changing speed to GEN3 data rate.
0	0h RW	**Receiver Use Preset Policy (RUPP):** Receiver Use Preset Policy(RUPP): This field applies to the Link Equalization Phase where the remote transmitter setting is being adjusted. When set, the received TS1 Use Preset bit will be checked. When clear, the Use Preset bit in the received TS1 will be ignored. Note: This bit must be set before changing speed to GEN3 data rate.

18.87 Remote Transmitter Preset Coefficient List 1 (RTPCL1) — Offset 454h

This register must be configured prior to enabling 8.0 GT/s data rate This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 454h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Preset/Coefficient Mode (PCM):** Preset/Coefficient Mode (PCM): This bit defines whether the Preset List or Coefficient List should be sent to the remote TX to adjust the remote TX setting. For Downstream Port, this is used in Phase 3 of the Link Equalization. For Upstream Port, this is used in Phase 2 of the Link Equalization. The list of coefficient or preset is configurable through the Remote Transmitter Preset Coefficient List [lb]1:4[rb] registers. When this bit is set, Coefficient Mode is enabled and the Remote Transmitter Preset Coefficient List [lb]1:4[rb] registers contain the Coefficient List. When this bit is clear, Preset Mode is enabled and the Remote Transmitter Preset Coefficient List [lb]1:3[rb] registers contain the Preset List.
30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Remote Transmitter Pre-Cursor Coefficient List 2/Preset List 4 (RTPRECL2PL4):** Remote Transmitter Pre-Cursor Coefficient List 2/Preset List 4 (RTPRECL2PL4): For Downstream Port: This field defines the pre-cursor coefficient List 2 or Preset List 4 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the pre-cursor coefficient List 2 or Preset List 4 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient or preset, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient or preset is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient or preset in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit when EQCFG1.RTPCOE = 0. If EQCFG1.RTPCOE = 1, the definition of this field is described in the EQCFG1.RTPCOE description. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.
23:18	0h RW	**Remote Transmitter Post-Cursor Coefficient List 1/Preset List 3 (RTPOSTCL1PL3):** Remote Transmitter Post-Cursor Coefficient List 1/Preset List 3 (RTPOSTCL1PL3): For Downstream Port: This field defines the post-cursor coefficient List 1 or Preset List 3 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the post-cursor coefficient List 1 or Preset List 3 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient or preset, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient or preset is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient or preset in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit when EQCFG1.RTPCOE = 0. If EQCFG1.RTPCOE = 1, the definition of this field is described in the EQCFG1.RTPCOE description. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.
17:12	0h RW	**Remote Transmitter Pre-Cursor Coefficient List 1/Preset List 2 (RTPRECL1PL2):** Remote Transmitter Pre-Cursor Coefficient List 1/Preset List 2 (RTPRECL1PL2): For Downstream Port: This field defines the pre-cursor coefficient List 1 or Preset List 2 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the pre-cursor coefficient List 1 or Preset List 2 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient or preset, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient or preset is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient or preset in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit when EQCFG1.RTPCOE = 0. If EQCFG1.RTPCOE = 1, the definition of this field is described in the EQCFG1.RTPCOE description. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.

Bit Range	Default & Access	Field Name (ID): Description
11:6	0h RW	**Remote Transmitter Post-Cursor Coefficient List 0/Preset List 1 (RTPOSTCL0PL1):** Remote Transmitter Post-Cursor Coefficient List 0/Preset List 1 (RTPOSTCL0PL1): For Downstream Port: This field defines the post-cursor coefficient List 0 or Preset List 1 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the post-cursor coefficient List 0 or Preset List 1 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient or preset, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient or preset is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient or preset in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit when EQCFG1.RTPCOE = 0. If EQCFG1.RTPCOE = 1, the definition of this field is described in the EQCFG1.RTPCOE description. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.
5:0	0h RW	**Remote Transmitter Pre-Cursor Coefficient List 0/Preset List 0 (RTPRECL0PL0):** Remote Transmitter Pre-Cursor Coefficient List 0/Preset List 0 (RTPRECL0PL0): For Downstream Port: This field defines the pre-cursor coefficient List 0 or Preset List 0 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the pre-cursor coefficient List 0 or Preset List 0 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient or preset, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient or preset is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient or preset in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit when EQCFG1.RTPCOE = 0. If EQCFG1.RTPCOE = 1, the definition of this field is described in the EQCFG1.RTPCOE description. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.

18.88 Remote Transmitter Preset Coefficient List 2 (RTPCL2) — Offset 458h

This register must be configured prior to enabling 8.0 GT/s data rate This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 458h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Remote Transmitter Post-Cursor Coefficient List 4/Preset List 9 (RTPOSTCL4PL9):** Remote Transmitter Post-Cursor Coefficient List 4/Preset List 9(RTPOSTCL4PL9): For Downstream Port: This field defines the post-cursor coefficient List 4 or Preset List 9 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the post-cursor coefficient List 4 or Preset List 9 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient or preset, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient or preset is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient or preset in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.

Bit Range	Default & Access	Field Name (ID): Description
23:18	0h RW	**Remote Transmitter Pre-Cursor Coefficient List 4/Preset List 8 (RTPRECL4PL8):** Remote Transmitter Pre-Cursor Coefficient List 4/Preset List 8 (RTPRECL4PL8): For Downstream Port: This field defines the pre-cursor coefficient List 4 or Preset List 8 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the pre-cursor coefficient List 4 or Preset List 8 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient or preset, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient or preset is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient or preset in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.
17:12	0h RW	**Remote Transmitter Post-Cursor Coefficient List 3/Preset List 7 (RTPOSTCL3PL7):** Remote Transmitter Post-Cursor Coefficient List 3/Preset List 7 (RTPOSTCL3PL7): For Downstream Port: This field defines the post-cursor coefficient List 3 or Preset List 7 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the post-cursor coefficient List 3 or Preset List 7 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient or preset, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient or preset is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient or preset in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit when EQCFG1.RTPCOE = 0. If EQCFG1.RTPCOE = 1, the definition of this field is described in the EQCFG1.RTPCOE description. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.
11:6	0h RW	**Remote Transmitter Pre-Cursor Coefficient List 3/Preset List 6 (RTPRECL3PL6):** Remote Transmitter Pre-Cursor Coefficient List 3/Preset List 6 (RTPRECL3PL6): For Downstream Port: This field defines the pre-cursor coefficient List 3 or Preset List 6 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the pre-cursor coefficient List 3 or Preset List 6 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient or preset, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient or preset is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient or preset in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit when EQCFG1.RTPCOE = 0. If EQCFG1.RTPCOE = 1, the definition of this field is described in the EQCFG1.RTPCOE description. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.
5:0	0h RW	**Remote Transmitter Post-Cursor Coefficient List 2/Preset List 5 (RTPOSTCL2PL5):** Remote Transmitter Post-Cursor Coefficient List 2/Preset List 5 (RTPOSTCL2PL5): For Downstream Port: This field defines the post-cursor coefficient List 2 or Preset List 5 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the post-cursor coefficient List 2 or Preset List 5 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient or preset, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient or preset is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient or preset in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit when EQCFG1.RTPCOE = 0. If EQCFG1.RTPCOE = 1, the definition of this field is described in the EQCFG1.RTPCOE description. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.

18.89 Remote Transmitter Preset Coefficient List 3 (RTPCL3) — Offset 45Ch

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 45Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Remote Transmitter Pre-Cursor Coefficient List 7 (RTPRECL7):** Remote Transmitter Pre-Cursor Coefficient List 7 (RTPRECL7): For Downstream Port: This field defines the pre-cursor coefficient List 7 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the pre-cursor coefficient List 7 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.
23:18	0h RW	**Remote Transmitter Post-Cursor Coefficient List 6 (RTPOSTCL6):** Remote Transmitter Post-Cursor Coefficient List 6 (RTPOSTCL6): For Downstream Port: This field defines the pre-cursor coefficient List 6 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the pre-cursor coefficient List 6 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.
17:12	0h RW	**Remote Transmitter Pre-Cursor Coefficient List 6 (RTPRECL6):** Remote Transmitter Pre-Cursor Coefficient List 6 (RTPRECL6): For Downstream Port: This field defines the pre-cursor coefficient List 6 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the pre-cursor coefficient List 6 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.
11:6	0h RW	**Remote Transmitter Post-Cursor Coefficient List 5 (RTPOSTCL5):** Remote Transmitter Post-Cursor Coefficient List 5 (RTPOSTCL5): For Downstream Port: This field defines the pre-cursor coefficient List 5 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the pre-cursor coefficient List 5 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.
5:0	0h RW	**Remote Transmitter Pre-Cursor Coefficient List 5/Preset List 10 (RTPRECL5PL10):** Remote Transmitter Pre-Cursor Coefficient List 5/Preset List 10 (RTPRECL5PL10): For Downstream Port: This field defines the pre-cursor coefficient List 5 or Preset List 10 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the pre-cursor coefficient List 5 or Preset List 10 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient or preset, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient or preset is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient or preset in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.

18.90 Remote Transmitter Preset Coefficient List 4 (RTPCL4) — Offset 460h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 460h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Remote Transmitter Post-Cursor Coefficient List 9 (RTPOSTCL9):** Remote Transmitter Post-Cursor Coefficient List 9 (RTPOSTCL9): For Downstream Port: This field defines the pre-cursor coefficient List 9 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the pre-cursor coefficient List 9 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.
23:18	0h RW	**Remote Transmitter Pre-Cursor Coefficient List 9 (RTPRECL9):** Remote Transmitter Pre-Cursor Coefficient List 9 (RTPRECL9): For Downstream Port: This field defines the pre-cursor coefficient List 9 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the pre-cursor coefficient List 9 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.
17:12	0h RW	**Remote Transmitter Post-Cursor Coefficient List 8 (RTPOSTCL8):** Remote Transmitter Post-Cursor Coefficient List 8 (RTPOSTCL8): For Downstream Port: This field defines the pre-cursor coefficient List 8 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the pre-cursor coefficient List 8 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.
11:6	0h RW	**Remote Transmitter Pre-Cursor Coefficient List 8 (RTPRECL8):** Remote Transmitter Pre-Cursor Coefficient List 8 (RTPRECL8): For Downstream Port: This field defines the pre-cursor coefficient List 8 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the pre-cursor coefficient List 8 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.
5:0	0h RW	**Remote Transmitter Post-Cursor Coefficient List 7 (RTPOSTCL7):** Remote Transmitter Post-Cursor Coefficient List 7 (RTPOSTCL7): For Downstream Port: This field defines the pre-cursor coefficient List 7 to be used for the remote TX in Phase 3. For Upstream Port: This field defines the pre-cursor coefficient List 7 to be used for the remote TX in Phase 2. When the remote TX has successfully applied the coefficient, the local RX will then perform Receiver Eye Width Margining to check the margin. The list of coefficient is traversed sequentially starting from List 0 to List N, where N is the most significant coefficient in the list. This field is treated as Preset or Coefficient based on RTPCL1.PCM bit. The value of coefficient programmed into this field assumes FS=64. Hardware will scale the coefficient accordingly based on the remote device FS value.

18.91 Figure Of Merit Status (FOMS) — Offset 464h

This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 464h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved
30:29	0h RW	**I:** Index (I): The FOMV field will reflect the Figure of Merit Scoreboard value for the index specified by this field. List N below refers to the Figure of Merit values captured in the scoreboard corresponding to the Preset or Coefficient List N. 00b: Index 0 =[gt] {List 2, List 1, List 0}. 01b: Index 1 =[gt] {List 5, List 4, List 3}. 10b: Index 2 =[gt] {List 8, List 7, List 6}. 11b: Index 3 =[gt] {Rsvd, List 10, List 9}.
28:24	0h RW	**Lane Number (LN):** Lane Number (LN): The FOMV field will reflect the Figure of Merit Scoreboard value for the lane specified by this field. 00000b: Lane 0. 00001b: Lane 1. 00010b: Lane 2. 00011b: Lane 3. Others: Reserved.
23:0	0h RO/V	**Figure of Merit Scoreboard Value (FOMSV):** Figure of Merit Scoreboard Value (FOMSV): This field will reflect the Figure of Merit Scoreboard entries referenced by the Lane Number and Index field in this register. For example, when Index == 00b, this field will reflect the Figure of Merit values for Lane specified in Lane Number field and the encoding of this field is as shown below: 23:16: Figure of Merit for Preset/ Coefficient List 2. 15:8 : Figure of Merit for Preset/Coefficient List 1. 7:0 : Figure of Merit for Preset/Coefficient List 0. If the Receiver Eye Width margining completes with error, the value of Figure of Merit should reflect 0x00.

18.92 Hardware Autonomous Equalization Control (HAEQ) — Offset 468h

Size:32 bits

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 468h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Hardware Autonomous Preset/Coefficient Count Per-Iteration (HAPCCPI):** Hardware Autonomous Preset/Coefficient Count Per-Iteration(HAPCCPI): This field defines the number of Preset/Coefficient to be traversed for every iteration of Recovery Equalization. For the Linear Mode, EQCFG2.HAPCSB specifies the total number of Presets/Coefficients to be checked in total while this field specifies the number of Presets/Coefficients to be checked per-iteration of Recovery Equalization. Hardware will enter Recovery Equalization and check the number of Presets/Coefficients specified by this field. Once that is done, hardware will exit Recovery Equalization and trigger another entry to Recovery Equalization to check another set of Presets/Coefficients. This goes on until the total number of Presets/Coefficients are checked. For Nine-Tiles Mode, EQCFG2.NTIC specifies the number of 9-tiles iterations, which indirectly specifies the total number of Presets/Coefficients to be checked in total. Similar to Linear Mode, this field specifies the number of Presets/Coefficients to be checked per-iteration of Recovery Equalization. 0h: 1 Preset/Coefficient per-iteration. 1h: 2 Preset/ Coefficient per-iteration. 2h: 3 Preset/Coefficient per-iteration. ... 9h: 10 Preset/ Coefficient per-iteration. Ah: 11 Preset/Coefficient per-iteration. Others: Reserved.

Bit Range	Default & Access	Field Name (ID): Description
27:20	0h RW	**FOM Error Mask (FOMEM):** FOM Error Mask(FOMEM): The FOM error counter will be masked(thus ignoring the FOM error) for all the FOM values prior to the FOM value specified in this field. If this field is programmed to 00h, this mechanism is disabled. This bit must be configured before training to GEN3 data rate.
19	0h RW	**MAC FOM Control (MACFOMC):** MAC FOM Control(MACFOMC): When set, MAC controls the advancement of the FOM values completely while in the Link Equalization mode. For downstream port, this is done in Phase 3 and for upstream port, this is done in Phase 2 of the Link Equalization. The dwelling time for each of the FOM values are programmed through the remaining fields of this register. When enabled, the hardware will start in Speeding Mode, where it will instruct the PHY to increment the FOM value after the Speeding Latency specified by HAEQ.SL field. Once the FOM value matches HAEQ.SFOMFM, the hardware switches from Speeding Mode to Dwelling Mode. In Dwelling mode, the MAC will instruct PHY to increment the FOM value after the Dwelling Latency specified by HAEQ.DL field. This is done until the link equalization phase is completed. When cleared, PHY controls the advancement of the FOM values completely, during the Link Equalization mode. This bit must be configured before training to GEN3 data rate.
18:16	0h RW	**Speeding Latency (SL):** Speeding Latency(SL): Specifies the residency time for a particular FOM value in Speeding Mode. 000b: 192 ns. 001b: 256 ns. 010b: 512 ns. 011b: 1 us. 100b: 2 us. 101b: 4 us. 110b: 8 us. 111b: 16 us. This register is only applicable when HAEC.MACFOMC is set and must be configured before setting the HAEC.MACFOMC bit.
15:8	0h RW	**Dwelling Latency (DL):** Dwelling Latency(DL): Specifies the residency time for a particular FOM value in Dwelling Mode. 00h: 2 us. 01h: 4 us. 02h: 6 us. ... FFh: 512 us. This register is only applicable when HAEC.MACFOMC is set and must be configured before setting the HAEC.MACFOMC bit.
7:0	0h RW	**Starting FOM For Margining (SFOMFM):** Starting FOM For Margining(SFOMFM): Define the FOM where MAC switches from Speeding Mode to Dwelling Mode after hitting the programmed FOM value in Hardware Autonomous Preset/Coefficient mode. This register is only applicable when HAEC.MACFOMC is set and must be configured before setting the HAEC.MACFOMC bit.

18.93 Local Transmitter Coefficient Override 1 (LTCO1) — Offset 470h

Based on the Port Configuration setting, if the maximum link width of the port is less than x4, the upper lane bits are not used. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 470h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (RSVD):** Reserved
25	0h RW	**Lane 1 Transmitter Coefficient Override Enable (L1TCOE):** Lane 1 Transmitter Coefficient Override Enable (L1TCOE): When set, the transmitter coefficient override values LTPCO1.L1TPRECO and LTPCO1.L1TPOSTCO are used as the local transmitter coefficient value, and the coefficient requested by the remote device is ignored. BIOS must ensure that the corresponding LTPCO1.L1TPRECO and LTPCO1.L1TPOSTCO fields above are programmed correctly prior to setting this bit. BIOS must ensure that the link runs in either GEN1 or GEN2 when the registers are configured. Once the register programming are done, BIOS should set the Perform Equalization bit to '1, Target Link Speed register to 0011b and initiate Link Retrain by setting the Retrain Link bit to 1b.

Bit Range	Default & Access	Field Name (ID): Description
24	0h RW	**Lane 0 Transmitter Coefficient Override Enable (L0TCOE):** Lane 0 Transmitter Coefficient Override Enable (L0TCOE): When set, the transmitter coefficient override values LTPCO1.L0TPRECO and LTPCO1.L0TPOSTCO are used as the local transmitter coefficient value, and the coefficient requested by the remote device is ignored. BIOS must ensure that the corresponding LTPCO1.L0TPRECO and LTPCO1.L0TPOSTCO fields above are programmed correctly prior to setting this bit. BIOS must ensure that the link runs in either GEN1 or GEN2 when the registers are configured. Once the register programming are done, BIOS should set the Perform Equalization bit to '1, Target Link Speed register to 0011b and initiate Link Retrain by setting the Retrain Link bit to 1b.
23:18	0h RW	**Lane 1 Transmitter Post-Cursor Coefficient Override (L1TPOSTCO):** Lane 1 Transmitter Post-Cursor Coefficient Override (L1TPOSTCO): For Downstream Port: This field defines the post-cursor coefficient override values to be used for the local TX in Phase 2. For Upstream Port: This field defines the post-cursor coefficient override values to be used for the local TX in Phase 3. This override value is used only when LTPCO1.L1TCOE = 1.
17:12	0h RW	**Lane 1 Transmitter Pre-Cursor Coefficient Override (L1TPRECO):** Lane 1 Transmitter Pre-Cursor Coefficient Override (L1TPRECO): For Downstream Port: This field defines the pre-cursor coefficient override values to be used for the local TX in Phase 2. For Upstream Port: This field defines the pre-cursor coefficient override values to be used for the local TX in Phase 3. This override value is used only when LTPCO1.L1TCOE = 1.
11:6	0h RW	**Lane 0 Transmitter Post-Cursor Coefficient Override (L0TPOSTCO):** Lane 0 Transmitter Post-Cursor Coefficient Override (L0TPOSTCO): For Downstream Port: This field defines the post-cursor coefficient override values to be used for the local TX in Phase 2. For Upstream Port: This field defines the post-cursor coefficient override values to be used for the local TX in Phase 3. This override value is used only when LTPCO1.L0TCOE = 1.
5:0	0h RW	**Lane 0 Transmitter Pre-Cursor Coefficient Override (L0TPRECO):** Lane 0 Transmitter Pre-Cursor Coefficient Override (L0TPRECO): For Downstream Port: This field defines the pre-cursor coefficient override values to be used for the local TX in Phase 2. For Upstream Port: This field defines the pre-cursor coefficient override values to be used for the local TX in Phase 3. This override value is used only when LTPCO1.L0TCOE = 1.

18.94 Local Transmitter Coefficient Override 2 (LTCO2) — Offset 474h

Based on the Port Configuration setting, if the maximum link width of the port is less than x4, the upper lane bits are not used. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 474h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**Lane 3 Transmitter Coefficient Override Enable (L3TCOE):** Lane 3 Transmitter Coefficient Override Enable (L3TCOE): When set, the transmitter coefficient override values LTPCO2.L3TPRECO and LTPCO2.L3TPOSTCO are used as the local transmitter coefficient value, and the coefficient requested by the remote device is ignored. BIOS must ensure that the corresponding LTPCO2.L3TPRECO and LTPCO2.L3TPOSTCO fields above are programmed correctly prior to setting this bit. BIOS must ensure that the link runs in either GEN1 or GEN2 when the registers are configured. Once the register programming are done, BIOS should set the Perform Equalization bit to '1, Target Link Speed register to 0011b and initiate Link Retrain by setting the Retrain Link bit to 1b.
24	0h RW	**Lane 2 Transmitter Coefficient Override Enable (L2TCOE):** Lane 2 Transmitter Coefficient Override Enable (L2TCOE): When set, the transmitter coefficient override values LTPCO2.L2TPRECO and LTPCO2.L2TPOSTCO are used as the local transmitter coefficient value, and the coefficient requested by the remote device is ignored. BIOS must ensure that the corresponding LTPCO2.L2TPRECO and LTPCO2.L2TPOSTCO fields above are programmed correctly prior to setting this bit. BIOS must ensure that the link runs in either GEN1 or GEN2 when the registers are configured. Once the register programming are done, BIOS should set the Perform Equalization bit to '1, Target Link Speed register to 0011b and initiate Link Retrain by setting the Retrain Link bit to 1b.
23:18	0h RW	**Lane 3 Transmitter Post-Cursor Coefficient Override (L3TPOSTCO):** Lane 3 Transmitter Post-Cursor Coefficient Override (L3TPOSTCO): For Downstream Port: This field defines the post-cursor coefficient override values to be used for the local TX in Phase 2. For Upstream Port: This field defines the post-cursor coefficient override values to be used for the local TX in Phase 3. This override value is used only when LTPCO2.L3TCOE = 1.
17:12	0h RW	**Lane 3 Transmitter Pre-Cursor Coefficient Override (L3TPRECO):** Lane 3 Transmitter Pre-Cursor Coefficient Override (L3TPRECO): For Downstream Port: This field defines the pre-cursor coefficient override values to be used for the local TX in Phase 2. For Upstream Port: This field defines the pre-cursor coefficient override values to be used for the local TX in Phase 3. This override value is used only when LTPCO2.L3TCOE = 1
11:6	0h RW	**Lane 2 Transmitter Post-Cursor Coefficient Override (L2TPOSTCO):** Lane 2 Transmitter Post-Cursor Coefficient Override (L2TPOSTCO): For Downstream Port: This field defines the post-cursor coefficient override values to be used for the local TX in Phase 2. For Upstream Port: This field defines the post-cursor coefficient override values to be used for the local TX in Phase 3. This override value is used only when LTPCO2.L2TCOE = 1.
5:0	0h RW	**Lane 2 Transmitter Pre-Cursor Coefficient Override (L2TPRECO):** Lane 2 Transmitter Pre-Cursor Coefficient Override (L2TPRECO): For Downstream Port: This field defines the pre-cursor coefficient override values to be used for the local TX in Phase 2. For Upstream Port: This field defines the pre-cursor coefficient override values to be used for the local TX in Phase 3. This override value is used only when LTPCO2.L2TCOE = 1.

18.95 GEN3 L0s Control (G3L0SCTL) — Offset 478h

This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 478h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**Gen3 Active State L0s Preparation Latency (G3ASL0SPL):** Gen3 Active State L0s Preparation Latency (G3ASL0SPL) Determines how long the Link layer has to indicate IDLE before the link initialization and control logic enters L0s 00: 0 clocks (enter immediately) 01: 1 clock ... FF: 255 clocks The value of this register is only used if the Gen3 L0s Entry Idle Control register is set to [quote]11[/quote] and operating in Gen3 mode.
23:22	0h RW	**Gen3 L0s Entry Idle Control (G3L0SIC):** Gen3 L0s Entry Idle Control (G3L0SIC): 00 : Allow entry into L0s after the link has been idle for a period of time equal to of the received N_FTS total entry time (1/4 * N_FTS * 16) 01 : Allow entry into L0s after the link has been idle for for a period of time equal to of the received N_FTS total entry time (1/2 * N_FTS * 16) 10 : Allow entry after the link has been idle for for a period of time equal to the received N_FTS total entry time (N_FTS * 16) 11: Allow entry into L0s after the link has been idle for a period specified in the Gen3 Active State L0s Preparation Latency register. This register is only applied when operating in Gen3 mode.
21:16	0h RO	**Reserved (RSVD):** Reserved
15:8	0h RW	**Gen3 Unique Clock N_FTS (G3UCNFTS):** Gen3 Unique Clock N_FTS (G3UCNFTS): Number of Fast Training Sequence ordered sets required to be transmitted for a root port Receiver to exit L0s in a unique (non-common) clock configuration (LCTL.CCC=0) when operating in Gen3 mode. The N_FTS value is sent in TS1 and TS2 training sets during link training. 00: 0 FTS sets 01: 1 FTS set ... FF: 255 FTS sets Note: When operating in Mobile Express mode, the output of this field is not used to determine the number of FTS to be sent on TXL0s exit. Mobile Express does not support Fast Training Sequence. Instead, SYNC is used to achieve bit lock. However, the output of this field is still used in L0s Entry Idle Control registers to determine the L0s Entry Idle latency.
7:0	0h RW	**Gen3 Common Clock N_FTS (G3CCNFTS):** Gen3 Common Clock N_FTS (G3CCNFTS): Number of Fast Training Sequence ordered sets required to be transmitted for a root port Receiver to exit L0s in a common clock configuration (LCTL.CCC=1) when operating in Gen3 mode. The N_FTS value is sent in TS1 and TS2 training sets during link training. 00: 0 FTS sets 01: 1 FTS set ... FF: 255 FTS sets Note: When operating in Mobile Express mode, the output of this field is not used to determine the number of FTS to be sent on TXL0s exit. Mobile Express does not support Fast Training Sequence. Instead, SYNC is used to achieve bit lock. However, the output of this field is still used in L0s Entry Idle Control registers to determine the L0s Entry Idle latency.

18.96 Equalization Configuration 2 (EQCFG2) — Offset 47Ch

Size:32 bits This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 47Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**Nine-Tiles Iteration Count (NTIC):** Nine-Tiles Iteration Count(NTIC): This field specifies the number of iterations to perform the 9-tiles search. Each iteration involves evaluating the neighboring 9-tiles for the best Preset/Coefficient margin and then use the Preset/Coefficient as the centerpoint to identify and evaluate the next 9-tiles. 00h: 1 iteration. 01h: 2 iterations. 02h: 3 iterations. ... FFh: 256 iterations.
23	0h RW	**Equalization Margining Disable (EMD):** Equalization Margining Disable(EMD):When set, the Root Port will not request the PHY to perform Receiver Margining by asserting RxEqEval on each Preset/Coefficient list traversed. This allows the receiver to still measure the Bit Error Count without margining. When cleared, the Root Port will request the PHY to perform Receiver Margining by asserting RxEqEval. This field is only valid when operating in Hardware Autonomous Preset/Coefficient Search mode. The Preset/Coefficient list will still be traversed to the end.

Bit Range	Default & Access	Field Name (ID): Description
22:20	0h RW	**Nine-Tiles Step Size (NTSS):** Nine-Tiles Step Size(NTSS): This field specifies the step size used to identify the surrounding 9-tiles to be used for margining. 000b: 1 step. 001b: 2 steps. 010b: 3 steps. 011b: 4 steps. 100b: 5 steps. 101b: 6 steps. 110b: 7 steps. 111b: 8 steps. Each of the steps is measured in terms of incrementing of decrementing the coefficient values.
19:16	0h RW	**Preset/Coefficient Evaluation Timeout (PCET):** Preset/Coefficient Evaluation Timeout(PCET): This field specifies the evaluation timeout for a single Preset/ Coefficient in the List when operating in Hardware Autonomous Preset/Coefficient Search mode. By spec, the evaluation phase must be completed before the 24 ms timeout. To support 12 Presets (11 Presets + 1 final good Preset), each Preset will have up to 2 ms for evaluation. This field allows the 2 ms timer to be programmable. This is useful if the EQCFG2.HAPCSB limits the Preset/Coefficient List to smaller than 12 such that each Preset/Coefficient could be evaluated for a time longer than 2 ms. 0h: 2 ms. 1h: 2.5 ms. 2h: 3 ms. 3h: 3.5 ms. 4h: 4 ms. 5h: 4.5 ms. 6h: 5 ms. 7h: 6 ms. 8h: 7 ms. 9h: 8 ms. Ah: 9 ms. Bh:10 ms. Ch:11 ms. Dh:21 ms. Eh:22 ms. Fh:23 ms.
15:12	0h RW	**Hardware Autonomous Preset/Coefficient Search Bound (HAPCSB):** Hardware Autonomous Preset/Coefficient Search Bound(HAPCSB): This field defines the number of Preset/Coefficient List to be traversed, out of 11 for Presets or out of 10 for Coefficients. The Preset/Coefficient list will be traversed from List 0 to the value specified by this field in incremental order. This field allows equalization to be done with smaller set of Preset/Coefficient list and each of the Preset/Coefficient list could be run for a longer time. 0h: Preset/Coefficient List 0 only. 1h: Preset/Coefficient List 0 - 1. 2h: Preset/Coefficient List 0 - 2. : : 9h: Preset/Coefficient List 0 - 9. Ah: Preset List 0 - 10/Coefficient List 0-9. Others: Reserved.
11	0h RW	**Nine-Tiles Equalization Mechanism Enable (NTEME):** Nine-Tiles Equalization Mechanism Enable(NTEME): When set, the Nine-Tiles Equalization Mechanism is enabled when running in Hardware Autonomous Preset/Coefficient Search Mode.
10	0h RW	**Mid-Point Equalization Mechanism Enable (MPEME):** Mid-Point Equalization Mechanism Enable(MPEME): When set, the Mid-Point Equalization Mechanism is enabled when running in Hardware Autonomous Preset/Coefficient Search Mode.
9:8	0h RW	**Receiver Eye Width Margin Error Threshold Multiplier (REWMETM):** Receiver Eye Width Margin Error Threshold Multiplier (REWMETM): This field specifies the multiplier to be used with REWMET field. 00b: Multiply REWMET by 1 (effectively no multiplier). 01b: Multiply REWMET by 10. 10b: Multiply REWMET by 100. 11b: Multiply REWMET by 1000.
7:0	0h RW	**Receiver Eye Width Margin Error Threshold (REWMET):** Receiver Eye Width Margin Error Threshold (REWMET): This field specifies the count threshold which upon exceeded, will cause controller to terminate the current iteration of Receiver Eye Width Margining and move on to the next preset or coefficient in the list. The value specified in this field will need to be multiplied with the multiplier specified in REWMETM field to get the final threshold values. 00h: Terminate on 1 x REWMETM errors. 01h: Terminate on 2 x REWMETM errors. 02h: Terminate on 4 x REWMETM errors. 03h: Terminate on 6 x REWMETM errors. : : FEh: Terminate on 508 x REWMETM errors. FFh: Never terminate. Rely on PHY to terminate the margining.

18.97 Monitor Mux (MM) — Offset 480h

This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 480h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO/V	**Monitor Signal State (MSST):** Monitor Signal State(MSST): The internal signal groupings selected by MM.MSS field is reflected in this field. The intention of this monitor signal is provide software a capability to monitor some of the GEN3 related parameters accumulated by the controller through the Link Equalization that are too costly to be mapped to dedicated registers. Implementation MUST NEVER expose any security related information through this Monitor Mux.
7:0	0h RW	**Monitor Signal Select (MSS):** Monitor Signal Select(MSS): This field is essentially the mux select for the Monitor Signal mux. Setting this field allows different monitor signals to be muxed out and readable by software through the MM.MSST field.

18.98 Lane0 P0 and P1 Preset-Coefficient Mapping (L0P0P1PCM) — Offset 500h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 500h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 1 Pre-Cursor Coefficient (P1PRECC):** Preset 1 Pre-Cursor Coefficient (P1PRECC): Pre-Cursor coefficient for Preset 1. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 1 Cursor Coefficient (P1CC):** Preset 1 Cursor Coefficient (P1CC): Cursor coefficient for Preset 1. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
17:12	0h RW	**Preset 0 Post-Cursor Coefficient (P0PSTCC):** Preset 0 Post-Cursor Coefficient (P0PSTCC): Post-Cursor coefficient for Preset 0. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 0 Pre-Cursor Coefficient (P0PRECC):** Preset 0 Pre-Cursor Coefficient (P0PRECC): Pre-Cursor coefficient for Preset 0. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 0 Cursor Coefficient (P0CC):** Preset 0 Cursor Coefficient (P0CC): Cursor coefficient for Preset 0. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.99 Lane0 P1, P2 and P3 Preset-Coefficient Mapping (L0P1P2P3PCM) — Offset 504h

This register must be configured prior to enabling 8.0 GT/s data rate This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 504h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 3 Cursor Coefficient (P3CC):** Preset 3 Cursor Coefficient (P3CC): Post-Cursor coefficient for Preset 3. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 2 Post-Cursor Coefficient (P2PSTCC):** Preset 2 Post-Cursor Coefficient (P2PSTCC): Post-Cursor coefficient for Preset 2. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
17:12	0h RW	**Preset 2 Pre-Cursor Coefficient (P2PRECC):** Preset 2 Pre-Cursor Coefficient (P2PRECC): Pre-Cursor coefficient for Preset 2. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 2 Cursor Coefficient (P2CC):** Preset 2 Cursor Coefficient (P2CC): Cursor coefficient for Preset 2. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 1 Post-Cursor Coefficient (P1PSTCC):** Preset 1 Post-Cursor Coefficient (P1PSTCC): Post-Cursor coefficient for Preset 1. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.100 Lane0 P3 and P4 Preset-Coefficient Mapping (L0P3P4PCM) — Offset 508h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 508h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 4 Post-Cursor Coefficient (P4PSTCC):** Preset 4 Post-Cursor Coefficient (P4PSTCC): Post-Cursor coefficient for Preset 4. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 4 Pre-Cursor Coefficient (P4PRECC):** Preset 4 Pre-Cursor Coefficient (P4PRECC): Pre-Cursor coefficient for Preset 4. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

Bit Range	Default & Access	Field Name (ID): Description
17:12	0h RW	**Preset 4 Cursor Coefficient (P4CC):** Preset 4 Cursor Coefficient (P4CC): Post-Cursor coefficient for Preset 4. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 3 Post-Cursor Coefficient (P3PSTCC):** Preset 3 Post-Cursor Coefficient (P3PSTCC): Post-Cursor coefficient for Preset 3. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 3 Pre-Cursor Coefficient (P3PRECC):** Preset 3 Pre-Cursor Coefficient (P3PRECC): Pre-Cursor coefficient for Preset 3. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.101 Lane0 P5 and P6 Preset-Coefficient Mapping (L0P5P6PCM) — Offset 50Ch

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 50Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 6 Pre-Cursor Coefficient (P6PRECC):** Preset 6 Pre-Cursor Coefficient (P6PRECC): Pre-Cursor coefficient for Preset 6. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 6 Cursor Coefficient (P6CC):** Preset 6 Cursor Coefficient (P6CC): Cursor coefficient for Preset 6. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
17:12	0h RW	**Preset 5 Post-Cursor Coefficient (P5PSTCC):** Preset 5 Post-Cursor Coefficient (P5PSTCC): Post-Cursor coefficient for Preset 5. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 5 Pre-Cursor Coefficient (P5PRECC):** Preset 5 Pre-Cursor Coefficient (P5PRECC): Pre-Cursor coefficient for Preset 5. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 5 Cursor Coefficient (P5CC):** Preset 5 Cursor Coefficient (P5CC): Cursor coefficient for Preset 5. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.102 Lane0 P6, P7 and P8 Preset-Coefficient Mapping (L0P6P7P8PCM) — Offset 510h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 510h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 8 Cursor Coefficient (P8CC):** Preset 8 Cursor Coefficient (P8CC): Post-Cursor coefficient for Preset 8. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 7 Post-Cursor Coefficient (P7PSTCC):** Preset 7 Post-Cursor Coefficient (P7PSTCC): Post-Cursor coefficient for Preset 7. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
17:12	0h RW	**Preset 7 Pre-Cursor Coefficient (P7PRECC):** Preset 7 Pre-Cursor Coefficient (P7PRECC): Pre-Cursor coefficient for Preset 7. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 7 Cursor Coefficient (P7CC):** Preset 7 Cursor Coefficient (P7CC): Cursor coefficient for Preset 7. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 6 Post-Cursor Coefficient (P6PSTCC):** Preset 6 Post-Cursor Coefficient (P6PSTCC): Post-Cursor coefficient for Preset 6. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.103 Lane0 P8 and P9 Preset-Coefficient Mapping (L0P8P9PCM) — Offset 514h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 514h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 9 Post-Cursor Coefficient (P9PSTCC):** Preset 9 Post-Cursor Coefficient (P9PSTCC): Post-Cursor coefficient for Preset 9. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 9 Pre-Cursor Coefficient (P9PRECC):** Preset 9 Pre-Cursor Coefficient (P9PRECC): Pre-Cursor coefficient for Preset 9. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

Bit Range	Default & Access	Field Name (ID): Description
17:12	0h RW	**Preset 9 Cursor Coefficient (P9CC):** Preset 9 Cursor Coefficient (P9CC): Post-Cursor coefficient for Preset 9. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 8 Post-Cursor Coefficient (P8PSTCC):** Preset 8 Post-Cursor Coefficient (P8PSTCC): Post-Cursor coefficient for Preset 8. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 8 Pre-Cursor Coefficient (P8PRECC):** Preset 8 Pre-Cursor Coefficient (P8PRECC): Pre-Cursor coefficient for Preset 8. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.104 Lane0 P10 Preset-Coefficient Mapping (L0P10PCM) — Offset 518h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 518h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Reserved (RSVD):** Reserved
17:12	0h RW	**Preset 10 Post-Cursor Coefficient (P10PSTCC):** Preset 10 Post-Cursor Coefficient (P10PSTCC): Post-Cursor coefficient for Preset 10. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 10 Pre-Cursor Coefficient (P10PRECC):** Preset 10 Pre-Cursor Coefficient (P10PRECC): Pre-Cursor coefficient for Preset 10. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 10 Cursor Coefficient (P10CC):** Preset 10 Cursor Coefficient (P10CC): Cursor coefficient for Preset 10. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.105 Lane0 LF and FS (L0LFFS) — Offset 51Ch

This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 51Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RO/V	**Remote PHY Full Swing Value (RPFSV):** Remote PHY Full Swing Value (RPFSV): The FS value for remote PHY. This value is logged from Lane 0.
23:22	0h RO	**Reserved (RSVD_1):** Reserved
21:16	0h RO/V	**Remote PHY Low Frequency Value (RPLFV):** Remote PHY Low Frequency Value (RPLFV): The LF value for remote PHY. This value is logged from Lane 0.
15:14	0h RO	**Reserved (RSVD_2):** Reserved
13:8	0h RW	**Local PHY Full Swing Value (LPFSV):** Local PHY Full Swing Value (LPFSV): The FS value for local PHY. Note: BIOS is responsible to ensure that the values programmed to the Preset-Coefficient Mapping table meets the requirement defined in PCI Express base specification with respect to FS value programmed in this field. This field must be configured prior to enabling 8.0 GT/s data rate. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
7:6	0h RO	**Reserved (RSVD_3):** Reserved
5:0	0h RW	**Local PHY Low Frequency Value (LPLFV):** Local PHY Low Frequency Value (LPLFV): The LF value for local PHY. Note: BIOS is responsible to ensure that the values programmed to the Preset-Coefficient Mapping table meets the requirement defined in PCI Express base specification with respect to LF value programmed in this field. This field must be configured prior to enabling 8.0 GT/s data rate. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.106 Lane1 P0 and P1 Preset-Coefficient Mapping (L1P0P1PCM) — Offset 520h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 520h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 1 Pre-Cursor Coefficient (P1PRECC):** Preset 1 Pre-Cursor Coefficient (P1PRECC): Pre-Cursor coefficient for Preset 1. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 1 Cursor Coefficient (P1CC):** Preset 1 Cursor Coefficient (P1CC): Cursor coefficient for Preset 1. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

Bit Range	Default & Access	Field Name (ID): Description
17:12	0h RW	**Preset 0 Post-Cursor Coefficient (P0PSTCC):** Preset 0 Post-Cursor Coefficient (P0PSTCC): Post-Cursor coefficient for Preset 0. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 0 Pre-Cursor Coefficient (P0PRECC):** Preset 0 Pre-Cursor Coefficient (P0PRECC): Pre-Cursor coefficient for Preset 0. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 0 Cursor Coefficient (P0CC):** Preset 0 Cursor Coefficient (P0CC): Cursor coefficient for Preset 0. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.107 Lane1 P1, P2 and P3 Preset-Coefficient Mapping (L1P1P2P3PCM) — Offset 524h

This register must be configured prior to enabling 8.0 GT/s data rate This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 524h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 3 Cursor Coefficient (P3CC):** Preset 3 Cursor Coefficient (P3CC): Post-Cursor coefficient for Preset 3. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 2 Post-Cursor Coefficient (P2PSTCC):** Preset 2 Post-Cursor Coefficient (P2PSTCC): Post-Cursor coefficient for Preset 2. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
17:12	0h RW	**Preset 2 Pre-Cursor Coefficient (P2PRECC):** Preset 2 Pre-Cursor Coefficient (P2PRECC): Pre-Cursor coefficient for Preset 2. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 2 Cursor Coefficient (P2CC):** Preset 2 Cursor Coefficient (P2CC): Cursor coefficient for Preset 2. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 1 Post-Cursor Coefficient (P1PSTCC):** Preset 1 Post-Cursor Coefficient (P1PSTCC): Post-Cursor coefficient for Preset 1. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.108 Lane1 P3 and P4 Preset-Coefficient Mapping (L1P3P4PCM) — Offset 528h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 528h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 4 Post-Cursor Coefficient (P4PSTCC):** Preset 4 Post-Cursor Coefficient (P4PSTCC): Post-Cursor coefficient for Preset 4. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 4 Pre-Cursor Coefficient (P4PRECC):** Preset 4 Pre-Cursor Coefficient (P4PRECC): Pre-Cursor coefficient for Preset 4. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
17:12	0h RW	**Preset 4 Cursor Coefficient (P4CC):** Preset 4 Cursor Coefficient (P4CC): Post-Cursor coefficient for Preset 4. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 3 Post-Cursor Coefficient (P3PSTCC):** Preset 3 Post-Cursor Coefficient (P3PSTCC): Post-Cursor coefficient for Preset 3. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 3 Pre-Cursor Coefficient (P3PRECC):** Preset 3 Pre-Cursor Coefficient (P3PRECC): Pre-Cursor coefficient for Preset 3. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.109 Lane1 P5 and P6 Preset-Coefficient Mapping (L1P5P6PCM) — Offset 52Ch

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 52Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 6 Pre-Cursor Coefficient (P6PRECC):** Preset 6 Pre-Cursor Coefficient (P6PRECC): Pre-Cursor coefficient for Preset 6. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 6 Cursor Coefficient (P6CC):** Preset 6 Cursor Coefficient (P6CC): Cursor coefficient for Preset 6. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

Bit Range	Default & Access	Field Name (ID): Description
17:12	0h RW	**Preset 5 Post-Cursor Coefficient (P5PSTCC):** Preset 5 Post-Cursor Coefficient (P5PSTCC): Post-Cursor coefficient for Preset 5. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 5 Pre-Cursor Coefficient (P5PRECC):** Preset 5 Pre-Cursor Coefficient (P5PRECC): Pre-Cursor coefficient for Preset 5. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 5 Cursor Coefficient (P5CC):** Preset 5 Cursor Coefficient (P5CC): Cursor coefficient for Preset 5. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.110 Lane1 P6, P7 and P8 Preset-Coefficient Mapping (L1P6P7P8PCM) — Offset 530h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 530h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 8 Cursor Coefficient (P8CC):** Preset 8 Cursor Coefficient (P8CC): Post-Cursor coefficient for Preset 8. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 7 Post-Cursor Coefficient (P7PSTCC):** Preset 7 Post-Cursor Coefficient (P7PSTCC): Post-Cursor coefficient for Preset 7. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
17:12	0h RW	**Preset 7 Pre-Cursor Coefficient (P7PRECC):** Preset 7 Pre-Cursor Coefficient (P7PRECC): Pre-Cursor coefficient for Preset 7. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 7 Cursor Coefficient (P7CC):** Preset 7 Cursor Coefficient (P7CC): Cursor coefficient for Preset 7. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 6 Post-Cursor Coefficient (P6PSTCC):** Preset 6 Post-Cursor Coefficient (P6PSTCC): Post-Cursor coefficient for Preset 6. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.111 Lane1 P8 and P9 Preset-Coefficient Mapping (L1P8P9PCM) — Offset 534h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 534h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 9 Post-Cursor Coefficient (P9PSTCC):** Preset 9 Post-Cursor Coefficient (P9PSTCC): Post-Cursor coefficient for Preset 9. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 9 Pre-Cursor Coefficient (P9PRECC):** Preset 9 Pre-Cursor Coefficient (P9PRECC): Pre-Cursor coefficient for Preset 9. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
17:12	0h RW	**Preset 9 Cursor Coefficient (P9CC):** Preset 9 Cursor Coefficient (P9CC): Post-Cursor coefficient for Preset 9. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 8 Post-Cursor Coefficient (P8PSTCC):** Preset 8 Post-Cursor Coefficient (P8PSTCC): Post-Cursor coefficient for Preset 8. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 8 Pre-Cursor Coefficient (P8PRECC):** Preset 8 Pre-Cursor Coefficient (P8PRECC): Pre-Cursor coefficient for Preset 8. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.112 Lane1 P10 Preset-Coefficient Mapping (L1P10PCM) — Offset 538h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 538h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Reserved (RSVD):** Reserved
17:12	0h RW	**Preset 10 Post-Cursor Coefficient (P10PSTCC):** Preset 10 Post-Cursor Coefficient (P10PSTCC): Post-Cursor coefficient for Preset 10. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 10 Pre-Cursor Coefficient (P10PRECC):** Preset 10 Pre-Cursor Coefficient (P10PRECC): Pre-Cursor coefficient for Preset 10. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

Bit Range	Default & Access	Field Name (ID): Description
5:0	0h RW	**Preset 10 Cursor Coefficient (P10CC):** Preset 10 Cursor Coefficient (P10CC): Cursor coefficient for Preset 10. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.113 Lane1 LF and FS (L1LFFS) — Offset 53Ch

This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 53Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RO/V	**Remote PHY Full Swing Value (RPFSV):** Remote PHY Full Swing Value (RPFSV): The FS value for remote PHY. This value is logged from Lane 0.
23:22	0h RO	**Reserved (RSVD_1):** Reserved
21:16	0h RO/V	**Remote PHY Low Frequency Value (RPLFV):** Remote PHY Low Frequency Value (RPLFV): The LF value for remote PHY. This value is logged from Lane 0.
15:14	0h RO	**Reserved (RSVD_2):** Reserved
13:8	0h RW	**Local PHY Full Swing Value (LPFSV):** Local PHY Full Swing Value (LPFSV): The FS value for local PHY. Note: BIOS is responsible to ensure that the values programmed to the Preset-Coefficient Mapping table meets the requirement defined in PCI Express base specification with respect to FS value programmed in this field. This field must be configured prior to enabling 8.0 GT/s data rate. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
7:6	0h RO	**Reserved (RSVD_3):** Reserved
5:0	0h RW	**Local PHY Low Frequency Value (LPLFV):** Local PHY Low Frequency Value (LPLFV): The LF value for local PHY. Note: BIOS is responsible to ensure that the values programmed to the Preset-Coefficient Mapping table meets the requirement defined in PCI Express base specification with respect to LF value programmed in this field. This field must be configured prior to enabling 8.0 GT/s data rate. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.114 Lane2 P0 and P1 Preset-Coefficient Mapping (L2P0P1PCM) — Offset 540h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 540h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 1 Pre-Cursor Coefficient (P1PRECC):** Preset 1 Pre-Cursor Coefficient (P1PRECC): Pre-Cursor coefficient for Preset 1. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 1 Cursor Coefficient (P1CC):** Preset 1 Cursor Coefficient (P1CC): Cursor coefficient for Preset 1. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
17:12	0h RW	**Preset 0 Post-Cursor Coefficient (P0PSTCC):** Preset 0 Post-Cursor Coefficient (P0PSTCC): Post-Cursor coefficient for Preset 0. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 0 Pre-Cursor Coefficient (P0PRECC):** Preset 0 Pre-Cursor Coefficient (P0PRECC): Pre-Cursor coefficient for Preset 0. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 0 Cursor Coefficient (P0CC):** Preset 0 Cursor Coefficient (P0CC): Cursor coefficient for Preset 0. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.115 Lane2 P1, P2 and P3 Preset-Coefficient Mapping (L2P1P2P3PCM) — Offset 544h

This register must be configured prior to enabling 8.0 GT/s data rate This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 544h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 3 Cursor Coefficient (P3CC):** Preset 3 Cursor Coefficient (P3CC): Post-Cursor coefficient for Preset 3. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 2 Post-Cursor Coefficient (P2PSTCC):** Preset 2 Post-Cursor Coefficient (P2PSTCC): Post-Cursor coefficient for Preset 2. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

Bit Range	Default & Access	Field Name (ID): Description
17:12	0h RW	**Preset 2 Pre-Cursor Coefficient (P2PRECC):** Preset 2 Pre-Cursor Coefficient (P2PRECC): Pre-Cursor coefficient for Preset 2. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 2 Cursor Coefficient (P2CC):** Preset 2 Cursor Coefficient (P2CC): Cursor coefficient for Preset 2. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 1 Post-Cursor Coefficient (P1PSTCC):** Preset 1 Post-Cursor Coefficient (P1PSTCC): Post-Cursor coefficient for Preset 1. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.116 Lane2 P3 and P4 Preset-Coefficient Mapping (L2P3P4PCM) — Offset 548h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 548h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 4 Post-Cursor Coefficient (P4PSTCC):** Preset 4 Post-Cursor Coefficient (P4PSTCC): Post-Cursor coefficient for Preset 4. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 4 Pre-Cursor Coefficient (P4PRECC):** Preset 4 Pre-Cursor Coefficient (P4PRECC): Pre-Cursor coefficient for Preset 4. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
17:12	0h RW	**Preset 4 Cursor Coefficient (P4CC):** Preset 4 Cursor Coefficient (P4CC): Post-Cursor coefficient for Preset 4. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 3 Post-Cursor Coefficient (P3PSTCC):** Preset 3 Post-Cursor Coefficient (P3PSTCC): Post-Cursor coefficient for Preset 3. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 3 Pre-Cursor Coefficient (P3PRECC):** Preset 3 Pre-Cursor Coefficient (P3PRECC): Pre-Cursor coefficient for Preset 3. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.117 Lane2 P5 and P6 Preset-Coefficient Mapping (L2P5P6PCM) — Offset 54Ch

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 54Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 6 Pre-Cursor Coefficient (P6PRECC):** Preset 6 Pre-Cursor Coefficient (P6PRECC): Pre-Cursor coefficient for Preset 6. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 6 Cursor Coefficient (P6CC):** Preset 6 Cursor Coefficient (P6CC): Cursor coefficient for Preset 6. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
17:12	0h RW	**Preset 5 Post-Cursor Coefficient (P5PSTCC):** Preset 5 Post-Cursor Coefficient (P5PSTCC): Post-Cursor coefficient for Preset 5. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 5 Pre-Cursor Coefficient (P5PRECC):** Preset 5 Pre-Cursor Coefficient (P5PRECC): Pre-Cursor coefficient for Preset 5. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 5 Cursor Coefficient (P5CC):** Preset 5 Cursor Coefficient (P5CC): Cursor coefficient for Preset 5. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.118 Lane2 P6, P7 and P8 Preset-Coefficient Mapping (L2P6P7P8PCM) — Offset 550h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 550h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 8 Cursor Coefficient (P8CC):** Preset 8 Cursor Coefficient (P8CC): Post-Cursor coefficient for Preset 8. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 7 Post-Cursor Coefficient (P7PSTCC):** Preset 7 Post-Cursor Coefficient (P7PSTCC): Post-Cursor coefficient for Preset 7. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

Bit Range	Default & Access	Field Name (ID): Description
17:12	0h RW	**Preset 7 Pre-Cursor Coefficient (P7PRECC):** Preset 7 Pre-Cursor Coefficient (P7PRECC): Pre-Cursor coefficient for Preset 7. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 7 Cursor Coefficient (P7CC):** Preset 7 Cursor Coefficient (P7CC): Cursor coefficient for Preset 7. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 6 Post-Cursor Coefficient (P6PSTCC):** Preset 6 Post-Cursor Coefficient (P6PSTCC): Post-Cursor coefficient for Preset 6. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.119 Lane2 P8 and P9 Preset-Coefficient Mapping (L2P8P9PCM) — Offset 554h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 554h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 9 Post-Cursor Coefficient (P9PSTCC):** Preset 9 Post-Cursor Coefficient (P9PSTCC): Post-Cursor coefficient for Preset 9. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 9 Pre-Cursor Coefficient (P9PRECC):** Preset 9 Pre-Cursor Coefficient (P9PRECC): Pre-Cursor coefficient for Preset 9. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
17:12	0h RW	**Preset 9 Cursor Coefficient (P9CC):** Preset 9 Cursor Coefficient (P9CC): Post-Cursor coefficient for Preset 9. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 8 Post-Cursor Coefficient (P8PSTCC):** Preset 8 Post-Cursor Coefficient (P8PSTCC): Post-Cursor coefficient for Preset 8. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 8 Pre-Cursor Coefficient (P8PRECC):** Preset 8 Pre-Cursor Coefficient (P8PRECC): Pre-Cursor coefficient for Preset 8. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.120 Lane2 P10 Preset-Coefficient Mapping (L2P10PCM) — Offset 558h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 558h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Reserved (RSVD):** Reserved
17:12	0h RW	**Preset 10 Post-Cursor Coefficient (P10PSTCC):** Preset 10 Post-Cursor Coefficient (P10PSTCC): Post-Cursor coefficient for Preset 10. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 10 Pre-Cursor Coefficient (P10PRECC):** Preset 10 Pre-Cursor Coefficient (P10PRECC): Pre-Cursor coefficient for Preset 10. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 10 Cursor Coefficient (P10CC):** Preset 10 Cursor Coefficient (P10CC): Cursor coefficient for Preset 10. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.121 Lane2 LF and FS (L2LFFS) — Offset 55Ch

This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 55Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RO/V	**Remote PHY Full Swing Value (RPFSV):** Remote PHY Full Swing Value (RPFSV): The FS value for remote PHY. This value is logged from Lane 0.
23:22	0h RO	**Reserved (RSVD_1):** Reserved
21:16	0h RO/V	**Remote PHY Low Frequency Value (RPLFV):** Remote PHY Low Frequency Value (RPLFV): The LF value for remote PHY. This value is logged from Lane 0.
15:14	0h RO	**Reserved (RSVD_2):** Reserved
13:8	0h RW	**Local PHY Full Swing Value (LPFSV):** Local PHY Full Swing Value (LPFSV): The FS value for local PHY. Note: BIOS is responsible to ensure that the values programmed to the Preset-Coefficient Mapping table meets the requirement defined in PCI Express base specification with respect to FS value programmed in this field. This field must be configured prior to enabling 8.0 GT/s data rate. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
7:6	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5:0	0h RW	**Local PHY Low Frequency Value (LPLFV):** Local PHY Low Frequency Value (LPLFV): The LF value for local PHY. Note: BIOS is responsible to ensure that the values programmed to the Preset-Coefficient Mapping table meets the requirement defined in PCI Express base specification with respect to LF value programmed in this field. This field must be configured prior to enabling 8.0 GT/s data rate. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.122 Lane3 P0 and P1 Preset-Coefficient Mapping (L3P0P1PCM) — Offset 560h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 560h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 1 Pre-Cursor Coefficient (P1PRECC):** Preset 1 Pre-Cursor Coefficient (P1PRECC): Pre-Cursor coefficient for Preset 1. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 1 Cursor Coefficient (P1CC):** Preset 1 Cursor Coefficient (P1CC): Cursor coefficient for Preset 1. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
17:12	0h RW	**Preset 0 Post-Cursor Coefficient (P0PSTCC):** Preset 0 Post-Cursor Coefficient (P0PSTCC): Post-Cursor coefficient for Preset 0. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 0 Pre-Cursor Coefficient (P0PRECC):** Preset 0 Pre-Cursor Coefficient (P0PRECC): Pre-Cursor coefficient for Preset 0. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 0 Cursor Coefficient (P0CC):** Preset 0 Cursor Coefficient (P0CC): Cursor coefficient for Preset 0. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.123 Lane3 P1, P2 and P3 Preset-Coefficient Mapping (L3P1P2P3PCM) — Offset 564h

This register must be configured prior to enabling 8.0 GT/s data rate This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 564h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 3 Cursor Coefficient (P3CC):** Preset 3 Cursor Coefficient (P3CC): Post-Cursor coefficient for Preset 3. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 2 Post-Cursor Coefficient (P2PSTCC):** Preset 2 Post-Cursor Coefficient (P2PSTCC): Post-Cursor coefficient for Preset 2. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
17:12	0h RW	**Preset 2 Pre-Cursor Coefficient (P2PRECC):** Preset 2 Pre-Cursor Coefficient (P2PRECC): Pre-Cursor coefficient for Preset 2. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 2 Cursor Coefficient (P2CC):** Preset 2 Cursor Coefficient (P2CC): Cursor coefficient for Preset 2. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 1 Post-Cursor Coefficient (P1PSTCC):** Preset 1 Post-Cursor Coefficient (P1PSTCC): Post-Cursor coefficient for Preset 1. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.124 Lane3 P3 and P4 Preset-Coefficient Mapping (L3P3P4PCM) — Offset 568h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 568h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 4 Post-Cursor Coefficient (P4PSTCC):** Preset 4 Post-Cursor Coefficient (P4PSTCC): Post-Cursor coefficient for Preset 4. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 4 Pre-Cursor Coefficient (P4PRECC):** Preset 4 Pre-Cursor Coefficient (P4PRECC): Pre-Cursor coefficient for Preset 4. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

Bit Range	Default & Access	Field Name (ID): Description
17:12	0h RW	**Preset 4 Cursor Coefficient (P4CC):** Preset 4 Cursor Coefficient (P4CC): Post-Cursor coefficient for Preset 4. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 3 Post-Cursor Coefficient (P3PSTCC):** Preset 3 Post-Cursor Coefficient (P3PSTCC): Post-Cursor coefficient for Preset 3. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 3 Pre-Cursor Coefficient (P3PRECC):** Preset 3 Pre-Cursor Coefficient (P3PRECC): Pre-Cursor coefficient for Preset 3. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.125 Lane3 P5 and P6 Preset-Coefficient Mapping (L3P5P6PCM) — Offset 56Ch

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 56Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 6 Pre-Cursor Coefficient (P6PRECC):** Preset 6 Pre-Cursor Coefficient (P6PRECC): Pre-Cursor coefficient for Preset 6. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 6 Cursor Coefficient (P6CC):** Preset 6 Cursor Coefficient (P6CC): Cursor coefficient for Preset 6. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
17:12	0h RW	**Preset 5 Post-Cursor Coefficient (P5PSTCC):** Preset 5 Post-Cursor Coefficient (P5PSTCC): Post-Cursor coefficient for Preset 5. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 5 Pre-Cursor Coefficient (P5PRECC):** Preset 5 Pre-Cursor Coefficient (P5PRECC): Pre-Cursor coefficient for Preset 5. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 5 Cursor Coefficient (P5CC):** Preset 5 Cursor Coefficient (P5CC): Cursor coefficient for Preset 5. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.126 Lane3 P6, P7 and P8 Preset-Coefficient Mapping (L3P6P7P8PCM) — Offset 570h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 570h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 8 Cursor Coefficient (P8CC):** Preset 8 Cursor Coefficient (P8CC): Post-Cursor coefficient for Preset 8. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 7 Post-Cursor Coefficient (P7PSTCC):** Preset 7 Post-Cursor Coefficient (P7PSTCC): Post-Cursor coefficient for Preset 7. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
17:12	0h RW	**Preset 7 Pre-Cursor Coefficient (P7PRECC):** Preset 7 Pre-Cursor Coefficient (P7PRECC): Pre-Cursor coefficient for Preset 7. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 7 Cursor Coefficient (P7CC):** Preset 7 Cursor Coefficient (P7CC): Cursor coefficient for Preset 7. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 6 Post-Cursor Coefficient (P6PSTCC):** Preset 6 Post-Cursor Coefficient (P6PSTCC): Post-Cursor coefficient for Preset 6. The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.127 Lane3 P8 and P9 Preset-Coefficient Mapping (L3P8P9PCM) — Offset 574h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 574h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RW	**Preset 9 Post-Cursor Coefficient (P9PSTCC):** Preset 9 Post-Cursor Coefficient (P9PSTCC): Post-Cursor coefficient for Preset 9. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
23:18	0h RW	**Preset 9 Pre-Cursor Coefficient (P9PRECC):** Preset 9 Pre-Cursor Coefficient (P9PRECC): Pre-Cursor coefficient for Preset 9. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

Bit Range	Default & Access	Field Name (ID): Description
17:12	0h RW	**Preset 9 Cursor Coefficient (P9CC):** Preset 9 Cursor Coefficient (P9CC): Post-Cursor coefficient for Preset 9. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 8 Post-Cursor Coefficient (P8PSTCC):** Preset 8 Post-Cursor Coefficient (P8PSTCC): Post-Cursor coefficient for Preset 8. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 8 Pre-Cursor Coefficient (P8PRECC):** Preset 8 Pre-Cursor Coefficient (P8PRECC): Pre-Cursor coefficient for Preset 8. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.128 Lane3 P10 Preset-Coefficient Mapping (L3P10PCM) — Offset 578h

This register must be configured prior to enabling 8.0 GT/s data rate. This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 578h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Reserved (RSVD):** Reserved
17:12	0h RW	**Preset 10 Post-Cursor Coefficient (P10PSTCC):** Preset 10 Post-Cursor Coefficient (P10PSTCC): Post-Cursor coefficient for Preset 10. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
11:6	0h RW	**Preset 10 Pre-Cursor Coefficient (P10PRECC):** Preset 10 Pre-Cursor Coefficient (P10PRECC): Pre-Cursor coefficient for Preset 10. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
5:0	0h RW	**Preset 10 Cursor Coefficient (P10CC):** Preset 10 Cursor Coefficient (P10CC): Cursor coefficient for Preset 10. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.129 Lane3 LF and FS (L3LFFS) — Offset 57Ch

This register is not applicable when operating in Mobile Express mode.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:19, F:X] + 57Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:24	0h RO/V	**Remote PHY Full Swing Value (RPFSV):** Remote PHY Full Swing Value (RPFSV): The FS value for remote PHY. This value is logged from Lane 0.
23:22	0h RO	**Reserved (RSVD_1):** Reserved
21:16	0h RO/V	**Remote PHY Low Frequency Value (RPLFV):** Remote PHY Low Frequency Value (RPLFV): The LF value for remote PHY. This value is logged from Lane 0.
15:14	0h RO	**Reserved (RSVD_2):** Reserved
13:8	0h RW	**Local PHY Full Swing Value (LPFSV):** Local PHY Full Swing Value (LPFSV): The FS value for local PHY. Note: BIOS is responsible to ensure that the values programmed to the Preset-Coefficient Mapping table meets the requirement defined in PCI Express base specification with respect to FS value programmed in this field. This field must be configured prior to enabling 8.0 GT/s data rate. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.
7:6	0h RO	**Reserved (RSVD_3):** Reserved
5:0	0h RW	**Local PHY Low Frequency Value (LPLFV):** Local PHY Low Frequency Value (LPLFV): The LF value for local PHY. Note: BIOS is responsible to ensure that the values programmed to the Preset-Coefficient Mapping table meets the requirement defined in PCI Express base specification with respect to LF value programmed in this field. This field must be configured prior to enabling 8.0 GT/s data rate. Note: The default value of this field is modified by the hardware to reflect the value queried from local PHY when Hardware Preset to Coefficient Mapping is completed.

18.130 Common Control (CC)—Offset 0h

FIA Private Configuration Register: Offset 00h: CC: Common Control

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/O	**Secured Register Lock (SRL):** Secured Register Lock (SRL): When this bit is set, all the secured registers will be locked and will be Read-Only. The following fields are locked by CC.SRL: - DRCRMx.
30:19	0h RO	**Reserved (RSVD):** Reserved (RSVD).
18	0h RW	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**Partition/Trunk Oscillator Clock Gating Enable (PTOCGE):** Partition/Trunk Oscillator Clock Gating Enable (PTOCGE): When set, the oscillator clock will be clock gated at the partition/trunk level when the conditions to clock gate are met. When clear, the oscillator clock will never be clock gated at the partition/trunk level. Note: If STRPFUSECFG.CDCGDIS is '1, hardware will always see '0 as an output from this register. BIOS reading this register should always return the correct value. Register Attribute: Static.
16	0h RW	**Oscillator/Side Clock Dynamic Clock Gating Enable (OSCDCGE):** Oscillator/Side Clock Dynamic Clock Gating Enable (OSCDCGE): When set, the oscillator and side clock will be dynamically clock gated when the conditions to clock gate are met. When clear, the oscillator and side clock will never be dynamically clock gated. Note: If STRPFUSECFG.CDCGDIS is '1, hardware will always see '0 as an output from this register. BIOS reading this register should always return the correct value. Register Attribute: Static.
15	0h RW	**Side Clock Partition/Trunk Clock Gating Enable (SCPTCGE):** Side Clock Partition/Trunk Clock Gating Enable (SCPTCGE): When set, the Side Clock will be clock gated at the partition/trunk level when the conditions to clock gate are met. When clear, the Side Clock will never be clock gated at the partition/trunk level. Note: If STRPFUSECFG1.CDCGDIS is '1, hardware will always see '0 as an output from this register. BIOS reading this register should always return the correct value. Register Attribute: Static.
14:2	0h RO	**Reserved (RSVD_2):** Reserved (RSVD).
1:0	0h RW	**IOSF Sideband ISM Idle Counter (SBIC):** IOSF Sideband ISM Idle Counter (SBIC): This register provides configuration flexibility to govern when the IOSF sideband ISM transitions to IDLE_REQ state with respect to sideband interface idle. 00b: IOSF sideband ISM will transition to IDLE_REQ after 16 clocks of idle on sideband interface. 01b: IOSF sideband ISM will transition to IDLE_REQ after 31 clocks of idle on sideband interface. 10b: IOSF sideband ISM will transition to IDLE_REQ after 64 clocks of idle on sideband interface. 11b: IOSF sideband ISM will never transition to IDLE_REQ on sideband interface. Register Attribute: Dynamic.

18.131 Device Reference Clock Request Mapping 1 (DRCRM1)—Offset 100h

Size:32 bits
FIA Private Configuration Register: Offset 100h: DRCRM1 - Device Reference Clock Request Mapping 1

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 76543210h

Bit Range	Default & Access	Field Name (ID): Description
31:28	7h RW/L	**Express Port 7 CLKREQ Mapping (P7CKRQM):** Express Port 7 CLKREQ Mapping (P7CKRQM): The mapping of Express Port 7 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 7 maps to CLKREQ0# pin. 0001b: Express Port 7 maps to CLKREQ1# pin. 0010b: Express Port 7 maps to CLKREQ2# pin. 0011b: Express Port 7 maps to CLKREQ3# pin. 0100b: Express Port 7 maps to CLKREQ4# pin. 0101b: Express Port 7 maps to CLKREQ5# pin. : : 1111b: Express Port 7 maps to CLKREQ15# pin. Software must never map multiple Express Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

Bit Range	Default & Access	Field Name (ID): Description
27:24	6h RW/L	**Express Port 6 CLKREQ Mapping (P6CKRQM):** Express Port 6 CLKREQ Mapping (P6CKRQM): The mapping of Express Port 6 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 6 maps to CLKREQ0# pin. 0001b: Express Port 6 maps to CLKREQ1# pin. 0010b: Express Port 6 maps to CLKREQ2# pin. 0011b: Express Port 6 maps to CLKREQ3# pin. 0100b: Express Port 6 maps to CLKREQ4# pin. 0101b: Express Port 6 maps to CLKREQ5# pin. : : 1111b: Express Port 6 maps to CLKREQ15# pin. Software must never map multiple Express Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

Bit Range	Default & Access	Field Name (ID): Description
23:20	5h RW/L	**Express Port 5 CLKREQ Mapping (P5CKRQM):** Express Port 5 CLKREQ Mapping (P5CKRQM): The mapping of Express Port 5 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 5 maps to CLKREQ0# pin. 0001b: Express Port 5 maps to CLKREQ1# pin. 0010b: Express Port 5 maps to CLKREQ2# pin. 0011b: Express Port 5 maps to CLKREQ3# pin. 0100b: Express Port 5 maps to CLKREQ4# pin. 0101b: Express Port 5 maps to CLKREQ5# pin. : : 1111b: Express Port 5 maps to CLKREQ15# pin. Software must never map multiple Express Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

Bit Range	Default & Access	Field Name (ID): Description
19:16	4h RW/L	**Express Port 4 CLKREQ Mapping (P4CKRQM):** Express Port 4 CLKREQ Mapping (P4CKRQM): The mapping of Express Port 4 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 4 maps to CLKREQ0# pin. 0001b: Express Port 4 maps to CLKREQ1# pin. 0010b: Express Port 4 maps to CLKREQ2# pin. 0011b: Express Port 4 maps to CLKREQ3# pin. 0100b: Express Port 4 maps to CLKREQ4# pin. 0101b: Express Port 4 maps to CLKREQ5# pin. : : 1111b: Express Port 4 maps to CLKREQ15# pin. Software must never map multiple Express Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

Bit Range	Default & Access	Field Name (ID): Description
15:12	3h RW/L	**Express Port 3 CLKREQ Mapping (P3CKRQM):** Express Port 3 CLKREQ Mapping (P3CKRQM): The mapping of Express Port 3 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 3 maps to CLKREQ0# pin. 0001b: Express Port 3 maps to CLKREQ1# pin. 0010b: Express Port 3 maps to CLKREQ2# pin. 0011b: Express Port 3 maps to CLKREQ3# pin. 0100b: Express Port 3 maps to CLKREQ4# pin. 0101b: Express Port 3 maps to CLKREQ5# pin. : : 1111b: Express Port 3 maps to CLKREQ15# pin. Software must never map multiple Express Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

Bit Range	Default & Access	Field Name (ID): Description
11:8	2h RW/L	**Express Port 2 CLKREQ Mapping (P2CKRQM):** Express Port 2 CLKREQ Mapping (P2CKRQM): The mapping of Port 2 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 2 maps to CLKREQ0# pin. 0001b: Express Port 2 maps to CLKREQ1# pin. 0010b: Express Port 2 maps to CLKREQ2# pin. 0011b: Express Port 2 maps to CLKREQ3# pin. 0100b: Express Port 2 maps to CLKREQ4# pin. 0101b: Express Port 2 maps to CLKREQ5# pin. : : 1111b: Express Port 2 maps to CLKREQ15# pin. Software must never map multiple Express Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

Bit Range	Default & Access	Field Name (ID): Description
7:4	1h RW/L	**Express Port 1 CLKREQ Mapping (P1CKRQM):** Express Port 1 CLKREQ Mapping (P1CKRQM): The mapping of Express Port 1 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 1 maps to CLKREQ0# pin. 0001b: Express Port 1 maps to CLKREQ1# pin. 0010b: Express Port 1 maps to CLKREQ2# pin. 0011b: Express Port 1 maps to CLKREQ3# pin. 0100b: Express Port 1 maps to CLKREQ4# pin. 0101b: Express Port 1 maps to CLKREQ5# pin. : : 1111b: Express Port 1 maps to CLKREQ15# pin. Software must never map multiple Express Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

Bit Range	Default & Access	Field Name (ID): Description
3:0	0h RW/L	**Express Port 0 CLKREQ Mapping (P0CKRQM):** Express Port 0 CLKREQ Mapping (P0CKRQM): The mapping of Express Port 0 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 0 maps to CLKREQ0# pin. 0001b: Express Port 0 maps to CLKREQ1# pin. 0010b: Express Port 0 maps to CLKREQ2# pin. 0011b: Express Port 0 maps to CLKREQ3# pin. 0100b: Express Port 0 maps to CLKREQ4# pin. 0101b: Express Port 0 maps to CLKREQ5# pin. : : 1111b: Express Port 0 maps to CLKREQ15# pin. Software must never map multiple Express Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

18.132 Device Reference Clock Request Mapping 2 (DRCRM2)—Offset 104h

Size:32 bits
FIA Private Configuration Register: Offset 104h: DRCRM2 - Device Reference Clock Request Mapping 2

Access Method

Type: MSG Register (Size: 32 bits)	**Device:** **Function:**

Default: FEDCBA98h

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RW/L	**Express Port 15 CLKREQ Mapping (P15CKRQM):** Express Port 15 CLKREQ Mapping (P15CKRQM): The mapping of Express Port 15 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 15 maps to CLKREQ0# pin. 0001b: Express Port 15 maps to CLKREQ1# pin. 0010b: Express Port 15 maps to CLKREQ2# pin. 0011b: Express Port 15 maps to CLKREQ3# pin. 0100b: Express Port 15 maps to CLKREQ4# pin. 0101b: Express Port 15 maps to CLKREQ5# pin. : : 1111b: Express Port 15 maps to CLKREQ15# pin. Software must never map multiple Express Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

Bit Range	Default & Access	Field Name (ID): Description
27:24	Eh RW/L	**Express Port 14 CLKREQ Mapping (P14CKRQM):** Express Port 14 CLKREQ Mapping (P14CKRQM): The mapping of Express Port 14 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 14 maps to CLKREQ0# pin. 0001b: Express Port 14 maps to CLKREQ1# pin. 0010b: Express Port 14 maps to CLKREQ2# pin. 0011b: Express Port 14 maps to CLKREQ3# pin. 0100b: Express Port 14 maps to CLKREQ4# pin. 0101b: Express Port 14 maps to CLKREQ5# pin. : : 1111b: Express Port 14 maps to CLKREQ15# pin. Software must never map multiple Express Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

Bit Range	Default & Access	Field Name (ID): Description
23:20	Dh RW/L	**Express Port 13 CLKREQ Mapping (P13CKRQM):** Express Port 13 CLKREQ Mapping (P13CKRQM): The mapping of Express Port 13 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 13 maps to CLKREQ0# pin. 0001b: Express Port 13 maps to CLKREQ1# pin. 0010b: Express Port 13 maps to CLKREQ2# pin. 0011b: Express Port 13 maps to CLKREQ3# pin. 0100b: Express Port 13 maps to CLKREQ4# pin. 0101b: Express Port 13 maps to CLKREQ5# pin. : : 1111b: Express Port 13 maps to CLKREQ15# pin. Software must never map multiple Express Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. Note: SPT-LP has only 12 PCIe ports. This field is Intel RW/L Reserved for SPT-LP and is only applicable to SPT-H. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

Bit Range	Default & Access	Field Name (ID): Description
19:16	Ch RW/L	**Express Port 12 CLKREQ Mapping (P12CKRQM):** Express Port 12 CLKREQ Mapping (P12CKRQM): The mapping of Express Port 12 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 12 maps to CLKREQ0# pin. 0001b: Express Port 12 maps to CLKREQ1# pin. 0010b: Express Port 12 maps to CLKREQ2# pin. 0011b: Express Port 12 maps to CLKREQ3# pin. 0100b: Express Port 12 maps to CLKREQ4# pin. 0101b: Express Port 12 maps to CLKREQ5# pin. : : 1111b: Express Port 12 maps to CLKREQ15# pin. Software must never map multiple Express Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

Bit Range	Default & Access	Field Name (ID): Description
15:12	Bh RW/L	**Express Port 11 CLKREQ Mapping (P11CKRQM):** Express Port 11 CLKREQ Mapping (P11CKRQM): The mapping of Express Port 11 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 11 maps to CLKREQ0# pin. 0001b: Express Port 11 maps to CLKREQ1# pin. 0010b: Express Port 11 maps to CLKREQ2# pin. 0011b: Express Port 11 maps to CLKREQ3# pin. 0100b: Express Port 11 maps to CLKREQ4# pin. 0101b: Express Port 11 maps to CLKREQ5# pin. : : 1111b: Express Port 11 maps to CLKREQ15# pin. Software must never map multiple Express Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

Bit Range	Default & Access	Field Name (ID): Description
11:8	Ah RW/L	**Express Port 10 CLKREQ Mapping (P10CKRQM):** Express Port 10 CLKREQ Mapping (P10CKRQM): The mapping of Express Port 10 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 10 maps to CLKREQ0# pin. 0001b: Express Port 10 maps to CLKREQ1# pin. 0010b: Express Port 10 maps to CLKREQ2# pin. 0011b: Express Port 10 maps to CLKREQ3# pin. 0100b: Express Port 10 maps to CLKREQ4# pin. 0101b: Express Port 10 maps to CLKREQ5# pin. : : 1111b: Express Port 10 maps to CLKREQ15# pin. Software must never map multiple Express Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

Bit Range	Default & Access	Field Name (ID): Description
7:4	9h RW/L	**Express Port 9 CLKREQ Mapping (P9CKRQM):** Express Port 9 CLKREQ Mapping (P9CKRQM): The mapping of Express Port 9 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 9 maps to CLKREQ0# pin. 0001b: Express Port 9 maps to CLKREQ1# pin. 0010b: Express Port 9 maps to CLKREQ2# pin. 0011b: Express Port 9 maps to CLKREQ3# pin. 0100b: Express Port 9 maps to CLKREQ4# pin. 0101b: Express Port 9 maps to CLKREQ5# pin. : : 1111b: Express Port 9 maps to CLKREQ15# pin. Software must never map multiple Express Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

Bit Range	Default & Access	Field Name (ID): Description
3:0	8h RW/L	**Express Port 8 CLKREQ Mapping (P8CKRQM):** Express Port 8 CLKREQ Mapping (P8CKRQM): The mapping of Express Port 8 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 8 maps to CLKREQ0# pin. 0001b: Express Port 8 maps to CLKREQ1# pin. 0010b: Express Port 8 maps to CLKREQ2# pin. 0011b: Express Port 8 maps to CLKREQ3# pin. 0100b: Express Port 8 maps to CLKREQ4# pin. 0101b: Express Port 8 maps to CLKREQ5# pin. : : 1111b: Express Port 8 maps to CLKREQ15# pin. PCIe Port 8 is not mapped to any CLKREQ# pins. Software must never map multiple PCIeExpress Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

18.133 Device Reference Clock Request Mapping 3 (DRCRM3)—Offset 108h

FIA Private Configuration Register: DRCRM3 - Device Reference Clock Request Mapping 3

Access Method

Type: MSG Register (Size: 32 bits)

Device:

Function:

Default: F000FFFFh

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RW/L	**GbE Port CLKREQ Mapping (GBEPCKRQM):** GbE Port CLKREQ Mapping (GBEPCKRQM): The mapping of GBE Port to the corresponding CLKREQ# pin is configured by this field. 0000b: GbE Port maps to CLKREQ0# pin. 0001b: GbE Port maps to CLKREQ1# pin. 0010b: GbE Port maps to CLKREQ2# pin. 0011b: GbE Port maps to CLKREQ3# pin. 0100b: GbE Port maps to CLKREQ4# pin. 0101b: GbE Port maps to CLKREQ5# pin. : : 1111b: GbE Port maps to CLKREQ15# pin. Software must never map multiple Express Ports and GbE Port to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.
27:16	0h RO	**Reserved (RSVD):** Reserved
15:12	Fh RW/L	**Express Port 19 CLKREQ Mapping (P19CKRQM):** Express Port 19 CLKREQ Mapping (P19CKRQM): The mapping of Express Port 19 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 19 maps to CLKREQ0# pin. 0001b: Express Port 19 maps to CLKREQ1# pin. 0010b: Express Port 19 maps to CLKREQ2# pin. 0011b: Express Port 19 maps to CLKREQ3# pin. 0100b: Express Port 19 maps to CLKREQ4# pin. 0101b: Express Port 19 maps to CLKREQ5# pin. : : 1111b: Express Port 19 maps to CLKREQ15# pin. Software must never map multiple Express Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

Bit Range	Default & Access	Field Name (ID): Description
11:8	Fh RW/L	**Express Port 18 CLKREQ Mapping (P18CKRQM):** Express Port 18 CLKREQ Mapping (P18CKRQM): The mapping of Express Port 18 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 18 maps to CLKREQ0# pin. 0001b: Express Port 18 maps to CLKREQ1# pin. 0010b: Express Port 18 maps to CLKREQ2# pin. 0011b: Express Port 18 maps to CLKREQ3# pin. 0100b: Express Port 18 maps to CLKREQ4# pin. 0101b: Express Port 18 maps to CLKREQ5# pin. : : 1111b: Express Port 18 maps to CLKREQ15# pin. Software must never map multiple Express Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

Bit Range	Default & Access	Field Name (ID): Description
7:4	Fh RW/L	**Express Port 17 CLKREQ Mapping (P17CKRQM):** Express Port 17 CLKREQ Mapping (P17CKRQM): The mapping of Express Port 17 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 17 maps to CLKREQ0# pin. 0001b: Express Port 17 maps to CLKREQ1# pin. 0010b: Express Port 17 maps to CLKREQ2# pin. 0011b: Express Port 17 maps to CLKREQ3# pin. 0100b: Express Port 17 maps to CLKREQ4# pin. 0101b: Express Port 17 maps to CLKREQ5# pin. : : 1111b: Express Port 17 maps to CLKREQ15# pin. Software must never map multiple Express Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

Bit Range	Default & Access	Field Name (ID): Description
3:0	Fh RW/L	**Express Port 16 CLKREQ Mapping (P16CKRQM):** Express Port 16 CLKREQ Mapping (P16CKRQM): The mapping of Express Port 16 to the corresponding CLKREQ# pin is configured by this field. 0000b: Express Port 16 maps to CLKREQ0# pin. 0001b: Express Port 16 maps to CLKREQ1# pin. 0010b: Express Port 16 maps to CLKREQ2# pin. 0011b: Express Port 16 maps to CLKREQ3# pin. 0100b: Express Port 16 maps to CLKREQ4# pin. 0101b: Express Port 16 maps to CLKREQ5# pin. : : 1111b: Express Port 16 maps to CLKREQ15# pin. Software must never map multiple Express Ports to the same CLKREQ# pin. The hardware behavior is undefined if this ever happens. Note: This field is must be configured prior to enabling any power management features. Note: For SPT-LP, there only CLKREQ[lb]0:5[rb]# are supported for PCIe. Configuring this field to map to any unsupported CLKREQ# pins may result in undefined hardware behavior. This register bit is Read-Only when the CC.SRL bit is set. Register Attribute: Static.

§ §

19 SATA Registers

This chapter documents the registers in Bus: 0, Device 18, Function 0.

19.1 ID — Offset 0h

Identifiers

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + 0h	8086 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:0	8086h RO	**Vendor ID (VID):** 16-bit field which indicates the company vendor as Intel.

19.2 Command (CMD) — Offset 4h

Command

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:18, F:0] + 4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:11	0h RO	**Reserved (RSVD):** Reserved
10	0h RW	**Interrupt Disable (ID):** This disables pin-based INTx# interrupts. This bit has no effect on MSI operation. When set, internal INTx# will not be generated. When cleared, internal INTx# are generated if there is an interrupt and MSI is not enabled.
9	0h RO	**Fast Back-to-Back Enable (FBE):** Reserved
8	0h RW	**SERR# Enable (SEE):** When set to 1, the HBA is allowed to generate SERR# on DPD or SATAGC.URD event that is enabled for SERR# generation. When cleared to 0, it is not.
7	0h RO	**Wait Cycle Enable (WCC):** Reserved
6	0h RW	**Parity Error Response Enable (PEE):** When set, the SATA Controller will corrupt the outbound DATA FIS CRC if a forwarded data parity error is indicated.
5	0h RO	**VGA:** Reserved

Bit Range	Default & Access	Field Name (ID): Description
4	0h RO	**Memory Write and Invalidate Enable (MWIE):** Reserved
3	0h RO	**Special Cycle Enable (SCE):** Reserved
2	0h RW	**Bus Master Enable (BME):** Controls the SATA Controller's ability to act as a master for data transfers. This bit does not impact the generation of completions for split transaction commands.
1	0h RW	**Memory Space Enable (MSE):** Controls access to the SATA Controller's target memory space (for AHCI).
0	0h RW	**I/O Space Enable (IOSE):** Controls access to the SATA Controller's target I/O space.

19.3 Device Status (STS) — Offset 6h

Device Status

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:18, F:0] + 6h	2B0 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C/V	**Detected Parity Error (DPE):** Set when the SATA Controller detects a parity error on its interface.
14	0h RW/1C/V	**Signalled System Error (SSE):** Set when SATA Controller generates an SERR#.
13	0h RW/1C/V	**Received Master-Abort Status (RMA):** Set when the SATA Controller receives a master abort to a cycle it generated.
12	0h RW/1C/V	**Received Target-Abort Status (RTA):** Set when the SATA Controller receives a target abort to a cycle it generated.
11	0h RW/1C/V	**Signalled Target-Abort Status (STA):** This bit must be set by a target device whenever it terminates a transaction with Target-Abort. Devices that will never signal Target-Abort do not need to implement this bit.
10:9	1h RO	**DEVSEL# Timing Status (DEVT):** Controls the device select time for the SATA Controller's PCI interface.
8	0h RW/1C/V	**Master Data Parity Error Detected (DPD):** Set when the SATA Controller, as a master, either detects a parity error or sees the parity error line asserted, and the parity error response bit (bit 6 of the command register) is set. This bit can only be set on read completions received from the backbone where there is a parity error.
7	1h RO	**Fast Back-to-Back Capable (FBC):** Reserved
6	0h RO	**Reserved (RSVD):** Reserved
5	1h RO	**66 MHz Capable (RSV):** Reserved
4	1h RO	**Capabilities List (CL):** Indicates the presence of a capabilities list. The minimum requirement for the capabilities list must be PCI power management for the SATA Controller.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO/V	**Interrupt Status (IS):** Reflects the state of INTx# messages, IRQ14 or IRQ15. This bit is set when the interrupt is to be asserted. This bit is a 0 after the interrupt is cleared
2:0	0h RO	**Reserved (RSVD):** Reserved

19.4 Revision ID (RID) — Offset 8h

Revision ID

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:18, F:0] + 8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Revision ID (RID):** Indicates stepping of the host controller hardware.

19.5 Programming Interface (PI) — Offset 9h

Programming Interface

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:18, F:0] + 9h	1 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	1h RO	**Interface (IF):** If CC.SCC=06h(AHCI mode), it indicates that this is an AHCI HBA that has a major revision of 1 (as specified in the AHCI Version register). If CC.SCC=04h(RAID mode), it indicates that there is no programming interface(IF=00h). Internally, under this condition, the SATA controller is in native mode and its I/O spaces are only accessible through the I/O BARs.

19.6 Class Code (CC) — Offset Ah

Class Code

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:18, F:0] + Ah	106 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	1h RO	**Base Class Code (BCC):** Indicates that this is a mass storage device.
7:0	6h RO	**Sub Class Code (SCC):** The value reported in this field is dependent on SATAGC.SMS and various fuses and configuration bits.

19.7 Cache Line Size (CLS) — Offset Ch

Cache Line Size

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:18, F:0] + Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Cache Line Size (CLS):** This register has no meaning for the SATA controller.

19.8 Master Latency Timer (MLT) — Offset Dh

Master Latency Timer

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:18, F:0] + Dh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Master Latency Timer (MLT):** This register has no meaning for the SATA controller.

19.9 Header Type (HTYPE) — Offset Eh

Header Type

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:18, F:0] + Eh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**Multi-function Device (MFD):** Indicates this controller is not part of a multi-function device.

Bit Range	Default & Access	Field Name (ID): Description
6:0	0h RO	**Header Layout (HL):** Indicates that the controller uses a target device layout.

19.10 MSI-X Table Base Address (MXTBA) — Offset 10h

This BAR is used to allocate 32K, 16K or 8K Memory space for the MSI-X Table. The Memory space size is determined by BIOS by making bit-14 and bit-13 Read-Only '1' or Read-Write '0' based on SATAGC.MSS[1:0].

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + 10h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RW	**BA:** Base address of memory space.
14	0h RW/V	**Base Address Bit 14 (BAB14):** When SATAGC.MSS[1:0]=00, this bit is Read Only '0' else it's Read Write '0'.
13	0h RW/V	**Base Address Bit 13 (BAB13):** When SATAGC.MSS[1:0]=00 or 01, this bit is Read Only '0' else it's Read Write '0'.
12:4	0h RO	**Reserved (RSVD):** Reserved
3	0h RO	**Prefetchable (PF):** Indicates that this range is not pre-fetchable.
2:1	0h RO	**Type (TP):** Indicates that this range can be mapped anywhere in 32-bit address space.
0	0h RO	**Resource Type Indicator (RTE):** Indicates a request for Memory space.

19.11 MXP Base Address (MXPBA) — Offset 14h

This BAR is used to allocate 256-byte Memory space for the MSI-X PBA.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + 14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**BA:** Base address of memory space (aligned to 256B).
7:4	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO	**Prefetchable (PF):** Indicates that this range is not pre-fetchable.
2:1	0h RO	**Type (TP):** Indicates that this range can be mapped anywhere in 32-bit address space.
0	0h RO	**Resource Type Indicator (RTE):** Indicates a request for Memory space.

19.12 SCMDBA — Offset 18h

BAR2 register define a 8-byte I/O space but this space is not implemented. Accesses to this space will get master abort. This space is created to workaround an issue in RST.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + 18h	1 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:3	0h RW	**Base Address (BAR):** Base address of the I/O space.
2:1	0h RO	**Reserved (RSVD):** Reserved
0	1h RO	**Resource Type Indicator (RTE):** Indicates a request for IO space.

19.13 SCTLBA — Offset 1Ch

BAR3 register define a 4-byte I/O space but this space is not implemented. Accesses to this space will get master abort. This space is created to workaround an issue in RST.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + 1Ch	1 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:2	0h RW	**Base Address (BAR):** Base address of the I/O space.
1	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
0	1h RO	**Resource Type Indicator (RTE):** Indicates a request for IO space.

19.14 AHCI Index Data Pair Base Address (AIDPBA) — Offset 20h

This BAR is used to allocate I/O space for the AHCI index/data pair mechanism. Note that hardware does not clear the BA bits (including BA4) when switching from IDE mode to non-IDE mode or vice versa. BIOS is responsible for clearing those bits to 0 since the number of writable bits changes after mode switching.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + 20h	1 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:5	0h RW	**BA:** Base address of the I/O space.
4:1	0h RO	**Reserved (RSVD):** Reserved
0	1h RO	**Resource Type Indicator (RTE):** Indicates a request for IO space.

19.15 AHCI Base Address (ABAR) — Offset 24h

This register represents a memory BAR allocating space for the AHCI memory registers. Note that bit[31:16] of this register must be programmed to a value greater than 0 to ensure the memory is mapped to an address of 1 MBytes and greater (i.e. ABAR must be 00010000h or greater). Otherwise, memory cycle targeting the ABAR range may not be accepted.. The Memory space size is determined by BIOS by making bit 15:11 Read-Only '1' or Read-Write '0' based on SATAGC.ASSEL[1:0].

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + 24h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RW	**BA:** Base address of register memory space.
18	0h RW	**Base Address Bit 18 (BAB18):** When SATAGC.ASSEL[2:0] selects an ABAR size bigger than 256K, this bit is Read Only '0' else it's Read Write '0'.

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**Base Address Bit 17 (BAB17):** When SATAGC.ASSEL[2:0] selects an ABAR size bigger than 128K, this bit is Read Only '0' else it's Read Write '0'.
16	0h RW	**Base Address Bit 16 (BAB16):** When SATAGC.ASSEL[2:0] selects an ABAR size bigger than 64K, this bit is Read Only '0' else it's Read Write '0'.
15	0h RW	**Base Address Bit 15 (BAB15):** When SATAGC.ASSEL[2:0] selects an ABAR size bigger than 32K, this bit is Read Only '0' else it's Read Write '0'.
14	0h RW	**Base Address Bit 14 (BAB14):** When SATAGC.ASSEL[2:0] selects an ABAR size bigger than 16K, this bit is Read Only '0' else it's Read Write '0'.
13:11	0h RW	**Base Address Bit 13-11 (BAB1311):** When SATAGC.ASSEL[2:0] selects an ABAR size bigger than 2K, this bit is Read Only '0' else it's Read Write '0'.
10:4	0h RO	**Reserved (RSVD):** Reserved
3	0h RO	**Prefetchable (PF):** Indicates that this range is not pre-fetchable
2:1	0h RO	**Type (TP):** Indicates that this range can be mapped anywhere in 32-bit address space
0	0h RO	**Resource Type Indicator (RTE):** Indicates a request for register memory space.

19.16 Sub System Identifiers (SS) — Offset 2Ch

This register is initialized to logic 0 by the assertion of PLTRST#. This register can be written only once after PLTRST# de-assertion. This register is not reset by FLR.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**Subsystem ID (SSID):** This is written by BIOS. No hardware action taken on this value.
15:0	0h RW/O	**Subsystem Vendor ID (SSVID):** This is written by BIOS. No hardware action taken on this value.

19.17 CAP — Offset 34h

Capabilities Pointer

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:18, F:0] + 34h	80 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	80h RW/L	**Capability Pointer (CP):** Indicates that the first capability pointer offset is offset 80h (the Message Signalled Interrupt capability). The following capability structures are linked by default: CAP.CP -> 80h (MSI) -> D0h (MSI-X)-> 70h (PCI Power) -> A8h (SATA) -> 00h end. BIOS may alter the capability structure list above (by programming a leading capability structure's Next Pointer field) if BIOS wants to bypass any specific capability. The RW/L register attribute allows for flexibility in determining the capability structure available in this PCI function. Refer to SATAGC.REGLOCK description in order to lock the register to become RO. This register is not reset by FLR.

19.18 Interrupt Information (INTR) — Offset 3Ch

Interrupt Information

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:18, F:0] + 3Ch	100 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	1h RW/O	**Interrupt Pin (IPIN):** This register tells which interrupt pin the device function uses. A value of 1 corresponds to INTA#. A value of 2 corresponds to INTB#. A value of 3 corresponds to INTC#. A value of 4 corresponds to INTD#. This register is not reset by FLR.
7:0	0h RW	**Interrupt Line (ILINE):** Software written value to indicate which interrupt line (vector) the interrupt is connected to. No hardware action is taken on this register. Interrupt Line register is not reset by FLR.

19.19 PCI Power Management Capability ID (PID) — Offset 70h

PCI Power Management Capability ID

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:18, F:0] + 70h	A801 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	A8h RW/L	**NEXT:** A8h is location of the Serial ATA Capability structure. The RW/L register attribute allows for flexibility in determining the capability structure available in this PCI function. Refer to SATAGC.REGLOCK description in order to lock the register to become RO. This register is not reset by FLR.
7:0	1h RO	**Cap ID (CID):** Indicates that this pointer is a PCI power management capability.

19.20 PC — Offset 72h

PCI Power Management Capabilities

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:18, F:0] + 72h	4003 h

Bit Range	Default & Access	Field Name (ID): Description
15:11	8h RO	**PME Support (PME_SUPPORT):** The default value is 01000 which indicates PME# can be generated from the D3HOT state in the SATA controller.
10	0h RO	**D2 Support (D2_SUPPORT):** The D2 state is not supported.
9	0h RO	**D1 Support (D1_SUPPORT):** The D1 state is not supported.
8:6	0h RO	**Aux Current (AUX_CURRENT):** PME# from D3COLD state is not supported, therefore this field is 000b.
5	0h RO	**Device Specific Initialization (DSI):** Indicates that no device-specific initialization is required.
4	0h RO	**Reserved (RSVD):** Reserved
3	0h RO	**PME Clock (PMEC):** Indicates that PCI clock is not required to generate PME#.
2:0	3h RO	**Version (VS):** Indicates support for Revision 1.2 of the PCI Power Management Specification.

19.21 PCI Power Management Control and Status (PMCS) — Offset 74h

PCI Power Management Control and Status

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:18, F:0] + 74h	8 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C/V	**PME Status (PMES):** This bit is set when a PME event is to be requested, and if this bit is set and PMEE is set, a PME# will be generated. This register field is not reset by FLR.
14:9	0h RO	**Reserved (RSVD):** Reserved
8	0h RW	**PME Enable (PMEE):** When set, the SATA controller generates PME# from D3HOT on a wake event. Note: Software is advised to clear PMEE together with PMES prior to changing CC.SCC thru SATAGC.SMS. This register field is not reset by FLR.
7:4	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
3	1h RO	**No Soft Reset (NSFRST):** A 1 indicates that devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset. Configuration context is preserved. Upon transition from the D3hot to the D0 state initialized state, no additional operating system intervention is required to preserve configuration context beyond writing to the PowerState bits. Regardless of this bit, the controller transition from D3hot to the D0 by a system or bus segment reset will return to the state D0 uninitialized with only PME context preserved if PME is supported and enabled.
2	0h RO	**Reserved (RSVD):** Reserved
1:0	0h RW	**Power State (PS):** This field is used both to determine the current power state of the SATA Controller and to set a new power state. The values are: 00 = D0 state; 11 = D3HOT state. When in the D3HOT state, the controller's configuration space is available, but the I/O and memory spaces are not. Additionally, interrupts are blocked. If software attempts to write a 10 or 01 to these bits, the write will be ignored. On software requirements in ensuring I/O space, memory space and Bus Master are disabled prior to entering D3 state.

19.22 Message Signaled Interrupt Identifier (MID) — Offset 80h

Message Signaled Interrupt Identifier

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:18, F:0] + 80h	7005 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	70h RW/L	**NEXT:** Indicates the next item in the list is the PCI power management pointer. This is the recommended value. BIOS may program this field to A8h indicating that the next item is Serial ATA Capability structure. The RW/L register attribute allows for flexibility in determining the capability structure available in this PCI function. Refer to SATAGC.REGLOCK description in order to lock the register to become RO. This register is not reset by FLR.
7:0	5h RO	**Capability ID (CID):** Capabilities ID indicates MSI.

19.23 Message Signalled Interrupt Message Control (MC) — Offset 82h

Message Signalled Interrupt Message Control

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:18, F:0] + 82h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**64-Bit Address Capable (C64):** Capable of generating a 32-bit message only.
6:4	0h RO	**Multiple Message Enable (MME):** When this field is cleared to 000 (and MSIE is set), only a single MSI message will be generated for all SATA ports, and bits [15:0] of the message vector will be driven from MD[15:0].
3:1	0h RO	**Multiple Message Capable (MMC):** Not supported.
0	0h RW	**MSI Enable (MSIE):** If set, MSI is enabled and traditional interrupt pins are not used to generate interrupts. Note that CMD.ID bit has not effect on MSI. Software must clear this bit to 0 to disable MSI first before changing the number of messages allocated in the MMC field.

19.24 Message Signalled Interrupt Message Address (MA) — Offset 84h

Message Signalled Interrupt Message Address

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + 84h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**ADDR:** Lower 32 bits of the system specified message address, always DWORD aligned.
1:0	0h RO	**Reserved (RSVD):** Reserved

19.25 Message Signalled Interrupt Message Data (MD) — Offset 88h

Message Signalled Interrupt Message Data

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:18, F:0] + 88h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW	**DATA:** This 16-bit field is programmed by system software if MSI is enabled. Its content is driven onto the lower word of the data bus of the MSI memory write transaction.

19.26 Port Mapping Register (MAP) — Offset 90h

Port Mapping Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + 90h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23	0h RW/O	**SATA Port 7 Disable (SPD7):** Similar to SPD0 but for port 7. This bit is RO-one if ANY of the fuse/strap/gpio condition below is true else this bit is RW-zero. PCIe/SATA muxing for port 7 based on fuses, soft straps and GPIO does not select SATA. This bit is only applicable to project(s): that has port 7 physically.
22	0h RW/O	**SATA Port 6 Disable (SPD6):** Similar to SPD0 but for port 6. This bit is RO-one if ANY of the fuse/strap/gpio condition below is true else this bit is RW-zero. PCIe/SATA muxing for port 6 based on fuses, soft straps and GPIO does not select SATA. This bit is only applicable to project(s): that has port 6 physically.
21	0h RW/O	**SATA Port 5 Disable (SPD5):** Similar to SPD0 but for port 5. This bit is RO-one if ANY of the fuse/strap/gpio condition below is true else this bit is RW-zero. PCIe/SATA muxing for port 5 based on fuses, soft straps and GPIO does not select SATA. This bit is only applicable to project(s): that has port 5 physically.
20	0h RW/O	**SATA Port 4 Disable (SPD4):** Similar to SPD0 but for port 4. This bit is RO-one if ANY of the fuse/strap/gpio condition below is true else this bit is RW-zero. PCIe/SATA muxing for port 4 based on fuses, soft straps and GPIO does not select SATA. This bit is only applicable to project(s): that has port 4 physically.
19	0h RW/O	**SATA Port 3 Disable (SPD3):** Similar to SPD0 but for port 3. This bit is RO-one if ANY of the fuse/strap/gpio condition below is true else this bit is RW-zero. 1.Fuse FFSATA7 (disable port 2 & 3). 2.Fuse FFSATA8 (disable port 1 & 3). 3. PCIe/SATA muxing for port 3 based on fuses, soft straps and GPIO does not select SATA. This bit is only applicable to project(s): that has port 3 physically.
18	0h RW/O	**SATA Port 2 Disable (SPD2):** Similar to SPD0 but for port 2. This bit is RO-one if ANY of the fuse/strap/gpio condition below is true else this bit is RW-zero. 1.Fuse FFSATA7 (disable port 2 & 3). 2.PCIe/SATA muxing for port 3 based on fuses, soft straps and GPIO does not select SATA. This bit is only applicable to project(s): that has port 3 physically.
17	0h RW/O	**SATA Port 1 Disable (SPD1):** Similar to SPD0 but for port 1. This bit is RO-one if ANY of the fuse/strap/gpio condition below is true else this bit is RW-zero. 1.Fuse FFSATA8 (disable port 1 & 3). 2.PCIe/SATA muxing for port 3 based on fuses, soft straps and GPIO does not select SATA. This bit is only applicable to project(s): that has port 1 physically.
16	0h RW/O	**SATA Port 0 Disable (SPD0):** A 1 prevents the SATA port from being enabled via config PCS.PxE. Write of 1 to PCS.PxE has no effect when the corresponding SPD[x] bit is 1.In preventing a port(s) from being enabled, BIOS shall first configure MAP.SPDx. And only then BIOS configures the PCS.PxE. This field is not reset by FLR. This bit is only applicable to project(s): that has port 0 physically.
15:8	0h RO	**Reserved (RSVD):** Reserved
7:0	0h RW	**Port Clock Disable (PCD):** When any of these bits is set to 1, the backbone clock driven to the associated port logic is gated and will not toggle. When this bit is cleared to 0, all clocks to the associated port logic will operate normally. Assignment of the bits is: Bit 7: Port 7, this bit is applicable to projects: that has port 7 physically; Bit 6: Port 6, this bit is only applicable to project(s): that has port 6 physically; Bit 5: Port 5, this bit is only applicable to project(s): that has port 5 physically; Bit 4: Port 4, this bit is only applicable to project(s): that has port 4 physically; Bit 3: Port 3, this bit is only applicable to project(s): that has port 3 physically; Bit 2: Port 2, this bit is only applicable to project(s): that has port 2 physically; Bit 1: Port 1, this bit is only applicable to project(s): that has port 1 physically; Bit 0: Port 0, this bit is only applicable to project(s): That has port 0 physically. If a particular port is not available, software shall set the corresponding bit to 1. Software can also set the corresponding bit(s) to 1 after disabling particular port(s). Software cannot set the PCD[port x]=1 if the corresponding config PCS.PxE=1 or AHCI MMIO GHC.PI[x]=1.

19.27 Port Control and Status (PCS) — Offset 94h

By default, the SATA ports are set (by hardware) to the disabled state (e.g. bits[5:0] == '0') as a result of an initial power on reset. When enabled by software, the ports can transition between the on, partial, and slumber states and can detect devices. When disabled, the port is in the off state and cannot detect any devices. This register is not reset by FLR. Note: AHCI specific notes: If an AHCI-aware or RAID enabled operating system is being booted then system BIOS shall insure that all supported SATA ports are enabled prior to passing control to the OS. Once the AHCI aware OS is booted it becomes the enabling/disabling policy owner for the individual SATA ports. This is accomplished by manipulating a port's PxSCTL.DET and PxCMD.SUD fields. Because an AHCI or RAID aware OS will typically not have knowledge of the PxE bits and because the PxE bits act as master on/off switches for the ports, pre-boot software must insure that these bits are set to '1' prior to booting the OS, regardless as to whether or not a device is currently on the port.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + 94h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23	0h RO/V	**Port 7 Present (P7P):** Same as P0P, except for port 7. This bit is only applicable to project(s): That has port 7 physically.
22	0h RO/V	**Port 6 Present (P6P):** Same as P0P, except for port 6. This bit is only applicable to project(s): That has port 6 physically.
21	0h RO/V	**Port 5 Present (P5P):** Same as P0P, except for port 5. This bit is only applicable to project(s): That has port 5 physically.
20	0h RO/V	**Port 4 Present (P4P):** Same as P0P, except for port 4. This bit is only applicable to project(s): That has port 4 physically.
19	0h RO/V	**Port 3 Present (P3P):** Same as P0P, except for port 3. This bit is only applicable to project(s): That has port 3 physically.
18	0h RO/V	**Port 2 Present (P2P):** Same as P0P, except for port 2. This bit is only applicable to project(s): That has port 2 physically.
17	0h RO/V	**Port 1 Present (P1P):** Same as P0P, except for port 1. This bit is only applicable to project(s): That has port 1 physically.
16	0h RO/V	**Port 0 Present (P0P):** When set, the SATA controller has detected the presence of a device on port 0. It may change at any time. Clearing P0E bit leads to clearing of this bit after implementation delay. Note: For system software that intends to clear all PCS.PxE bits that are previously 1 to 0 and then to 1 again in two consecutive write cycles, software shall poll on this bit being 0 before setting P0E bit to 1. This bit is only applicable to project(s): That has port 0 physically.
15:8	0h RO	**Reserved (RSVD):** Reserved
7	0h RW/V	**Port 7 Enabled (P7E):** When MAP.SPD[7] is 1, this bit is reserved and is read-only 0. Otherwise if none of the above is true, this field is RW and the definition of this bit is the same as P0E, except for port 7. This bit takes precedence over P7CMD.SUD. This bit is only applicable to project(s): That has port 7 physically.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW/V	**Port 6 Enabled (P6E):** When MAP.SPD[6] is 1, this bit is reserved and is read-only 0. Otherwise if none of the above is true, this field is RW and the definition of this bit is the same as P0E, except for port 6. This bit takes precedence over P6CMD.SUD. This bit is only applicable to project(s): That has port 6 physically.
5	0h RW/V	**Port 5 Enabled (P5E):** When MAP.SPD[5] is 1, this bit is reserved and is read-only 0. Otherwise if none of the above is true, this field is RW and the definition of this bit is the same as P0E, except for port 5. This bit takes precedence over P5CMD.SUD. This bit is only applicable to project(s): That has port 5 physically.
4	0h RW/V	**Port 4 Enabled (P4E):** When MAP.SPD[4] is 1, this bit is reserved and is read-only 0. Otherwise if none of the above is true, this bit is RW and same as P0E but for port 4 and takes precedence over P4CMD.SUD. This bit is only applicable to project(s): That has port 4 physically.
3	0h RW/V	**Port 3 Enabled (P3E):** When MAP.SPD[3] is 1, this bit is reserved and is read-only 0. Otherwise if none of the above is true, this bit is RW and same as P0E but for port 3 and takes precedence over P3CMD.SUD. This bit is only applicable to project(s): That has port 3 physically.
2	0h RW/V	**Port 2 Enabled (P2E):** When MAP.SPD[2] is 1, this bit is reserved and is read-only 0. Otherwise if none of the above is true, this bit is RW and same as P0E but for port 2 and takes precedence over P2CMD.SUD. This bit is only applicable to project(s): That has port 2 physically.
1	0h RW/V	**Port 1 Enabled (P1E):** When MAP.SPD[1] is 1, this bit is reserved and read-only 0. Otherwise if none of the above is true, this bit is RW and same as P0E but for port 1 and takes precedence over P1CMD.SUD. This bit is only applicable to project(s): That has port 1 physically.
0	0h RW/V	**Port 0 Enabled (P0E):** When MAP.SPD[0] is 1, this bit is reserved and read-only 0. When set, the port is enabled. When cleared, the port is disabled. When enabled, the port can transition between the on, partial, and slumber states and can detect devices. When disabled, the port is in the off state and cannot detect any devices. This bit takes precedence over P0CMD.SUD. Note: The recommendation for software code that intends to clear all PCS.PxE bits that are previously 1 to 0 and then to 1 again immediately shall refer to the polling requirement as described in P0P register bit. At any time that BIOS or software is clearing PCS.PxE from 1 to 0, due to time needed for port staggering hardware process (up to 6 ports) to complete, BIOS and software shall delay the write to set the TM.PCD register by 1.4us.

19.28 SATA General Configuration (SATAGC) — Offset 9Ch

SATA General Configuration

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + 9Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/O	**Register Lock (REGLOCK):** BIOS can set this bit to 1 to lock the following registers with RW/L attribute: CAP,CP, MID.NEXT, PIE.NEXT, SATACR0.NEXT. Once locked the register attribute of above list changes from RW/L to RO holding the existing value. BIOS is required to program this field to 1 prior to handle off to OS. If BIOS needs the SATA host controller to change operation a few times (i.e. changing CC.SCC mode) and need different capability structures for each specific operation mode, BIOS need not activate the lock until BIOS is ready to hand off to OS. BIOS may need to separate write access to this byte offset (x9Fh) from write to the lower 3-byte of the dword (x9C-9Eh) if there is a need to program the lower 3-byte dword location early during boot process. This field is not reset by FLR.
30:18	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**Do SERR Disable (DOSERRD):** When 1,SERR reporting is disabled (DO_SERR Sideband message sending is disabled or SERR# Wire is supressed. STS.SSE setting is not governed by this policy bit). When 0, SERR reporting is enabled (DO_SERR Sideband message sending is enabled or SERR# Wire is not supressed)
16	0h RW	**SATA Mode Select (SMS):** Software (SW) programs these bits to control the mode in which the SATA HBA should operate: 0b = AHCI mode; 1b = RAID mode; Notes : SW shall not manipulate SATAGC.SMS during runtime operation; i.e. the OS will not do this. The BIOS may choose to switch from one mode to another during POST; AHCI mode may be selected when RAID feature is enabled by fuse; RAID mode may only be selected when FFSATA5 & FFSATA3 (concatenated value not indicating No RAID); This register field is not reset by FLR.
15	0h RW	**Data Phase Parity Error Enable (DPPEE):** When 1, IOSF data phase parity error handling is enabled. When 0, the data phase parity error handling is disabled.
14:12	0h RW	**Write Request_Size Select/Max_Payload_Size (WRRSELMPS):** These two bits select the max write request size that SATA host controller will initiate for DMA write to memory. SATA host controller will internally break up larger write request based on these bits. The request is address-aligned to the selected size. Defined encodings for this field are: 000b = 128 address aligned bytes max payload size; 111b = 64 address aligned bytes max payload size. All other values are reserved for SATA host controller. This field is not reset by FLR.
11	0h RW	**Command Parity Error Enable (CPEE):** When 1, command parity error handling is enabled. When 0, the command parity error handling is disabled.
10	0h RW	**SATA Controller Function Disable (SCFD):** BIOS program this bit to 1 to disable the SATA Controller function. When 0, SATA Controller function is enabled. When disable, SATA Host Controller will not claimed the register access targeting its Configuration Space. In IOSF primary Fabric Decode scheme, it's expected BIOS also program the corresponding bit used by the Fabric Decoder accordingly hence both SATA SIP and Fabric Decoder are in sync, and BIOS need to program this bit before programming the one in Fabric Decoder. Once this bit is set, BIOS isnot able to revert it back to Function Enable until next round of platform reset.
9	0h RW	**Unsupported Request Reporting Enable (URRE):** If set to 1 by software, it allows reporting of an Unsupported Request as a system error. If both URRE and PCI configuration SERR# Enable registers are set to a 1, then the agent must set the Signalled System Error bit in the PCI Status register and send a DO_SERR message in IOSF-SB interface.
8	0h RW/1C/V	**Unsupported Request Detected (URD):** Set to 1 by hardware upon detecting an Unsupported Request on IOSF Primary interface that is not considered Advisory Non-Fatal. Cleared to 0 by SW.
7	0h RW/O	**Alternate ID Enable (AIE):** When programmed to 0, HW will report the following device id's: 2822h for desktop or 282Ah for mobile. When programmed to a 1, HW will not report these device id's. Note: Programming this bit to a 1 will prevent the Windows in-box version of the Intel AHCI driver from loading on the platform - will require that the user perform an 'F6' install of the Intel driver that is appropriate for the reported DID. This field is applicable when the AHCI is configured for RAID mode of operation. It has no impact for AHCI and IDE modes of operation. Note: BIOS is recommended to program this bit prior to programming the MAP.SMS field to reflect RAID. This field is reset by PLTRST# and BIOS is required to reprogram the value (either 0 or 1) after resuming from S3, S4 or S5. This insures that the value is properly and cannot be changed during runtime. This field is not reset by FLR.
6	0h RW/O/V	**AIE0 DevID Selection (DEVIDSEL):** This register allows BIOS to select Device ID when AIE=0 and Server Feature (SATA AIE DEVIDSEL) Disable Fuse =0. This bit only has effect in Desktop SKU. In Mobile SKU this bit has no effect at all. Refer to config register offset 09h PI for usage. Note: WBG BIOS is required to program this field to 1 together with the write to the AIE bit in a single configuration write cycle. BIOS is required to program this bit to 0 together with the write to the AIE bit in a single configuration write cycle. NOTE: When Server Feature (SATA AIE DEVIDSEL) Disable Fuse is programmed to 1, this disables the writeability of this DEVIDSEL register bit, and becomes RO with a value of 0, which only allows a choice of 2822h. This field is not reset by FLR.
5	0h RW/O	**FLR Capability Selection (FLRCSSEL):** This allows the FLR Capability to be bypassed. Refer to config offset B0h. BIOS is required to program this bit to 1 and config offset A8h SATACR0.NEXT to 00h. This field is not reset by FLR.

Bit Range	Default & Access	Field Name (ID): Description
4:3	0h RW/O	**MXTBA Size Select (MSS):** These 2 bits select the size of the Memory space for the MSI-X Table defined in BAR 0 (Configuration space offset 10h). MSI-X Table Memory space size is 32k when MSS[1:0]=00, 16k when MSS[1:0]=01, 8k when MSS[1:0]=10. This field is not reset by FLR.
2:0	0h RW/O	**ABAR Size Select (ASSEL):** These 3 bits select the size of the Memory space for the ABAR in BAR 5 (Configuration space offset 24h). ABAR Memory space size is 2k when ASSEL[2:0]=000, 16k when ASSEL[2:0]=001, 32k when ASSEL[2:0]=010, 64k when ASSEL[2:0]=011, 128k when ASSEL[2:0]=100, 256k when ASSEL[2:0]=101, 512k when ASSEL[2:0]=110. This field is not reset by FLR.

19.29 SATA Initialization Register Index (SIRI) — Offset A0h

SATA Initialization Register Index

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:18, F:0] + A0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:2	0h RW	**Index (IDX):** 6-bit index pointer into the 256-byte space. Data is written into the SIRD register and read from the SIRD register. This point to a DWord register. The byte enables on the SIRD register affect what will be written.
1:0	0h RO	**Reserved (RSVD):** Reserved

19.30 SATA Initialization Register Data (SIRD) — Offset A4h

SATA Initialization Register Data

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + A4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Data (DTA):** 32-bit data value that is written to the register pointed to by SIRI, or read from the register pointed to by SIRI.

19.31 Serial ATA Capability Register 0 (SATACR0) — Offset A8h

Note that the SATACR0.NEXT is not changed from RO to become RWO because there is an existing method (SATAGC.FLRCSSEL bit) to bypass the FLR Capability structure. And FLR Capability ID.NEXT is already indicating end of capability structure, it does not need change to be RWO.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + A8h	100012 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23:20	1h RO	**Major Revision (MAJREV):** Major revision number of the SATA Capability Pointer implemented.
19:16	0h RO	**Minor Revision (MINREV):** Minor revision number of the SATA Capability Pointer implemented.
15:8	0h RW/L	**NEXT:** 00h indicating the final item in the Capability List. The RW/L register attribute allows for flexibility in determining the capability structure available in this PCI function. Refer to SATAGC.REGLOCK description in order to lock the register to become RO. This register is not reset by FLR.
7:0	12h RO	**CAP:** The value of 12h has been assigned by the PCI SIG to designate the SATA Capability pointer.

19.32 Serial ATA Capability Register 1 (SATACR1) — Offset ACh

Serial ATA Capability Register 1

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + ACh	48 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:4	4h RO	**BAR Offset (BAROFST):** Indicates the offset into the BAR where the AHCI Index/Data pair are located (in Dword granularity). The Index and Data I/O registers are located at offset 10h within the I/O space defined by LBAR (BAR4). A value of 004h indicates offset 10h. 000h = 0h offset; 001h = 4h offset; 002h = 8h offset; 003h = Ch offset; 004h = 10h offset; ...; FFFh = 3FFFh offset (max 16 KB)
3:0	8h RO	**BAR Location (BARLOC):** Indicates the absolute PCI Configuration Register address of the BAR containing the Index/Data pair (in Dword granularity). The Index and Data I/O registers reside within the space defined by LBAR (BAR4) in the SATA controller. A value of 8h indicates offset 20h, which is LBAR (BAR4). 0000 - 0011b = reserved; 0100b = 10h => BAR0; 0101b = 14h => BAR1; 0110b = 18h => BAR2; 0111b = 1Ch => BAR3; 1000b = 20h => AIDPBA; 1001b = 24h => BAR5; 1010 - 1110b = reserved; 1111b = Index/Data pair in PCI Configuration space which is not supported.

19.33 FLR Capability ID (FLRCID) — Offset B0h

FLR Capability ID

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:18, F:0] + B0h	13 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	**NEXT:** 00h indicating the final item in the Capability List.
7:0	13h RO/V	**Capability ID (CID):** The value of this field depends on the FLRCSSEL bit. • SATAGC.FLRCSSEL = 0, Capability ID = 13h. • SATAGC.FLRCSSEL = 1, Capability ID = 00h (capability is bypassed).

19.34 FLR Capability Length and Version (FLRCAP) — Offset B2h

This register shall be read-only 0 when SATAGC.FLRCSSEL=1. This register is not reset by FLR.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:18, F:0] + B2h	306 h

Bit Range	Default & Access	Field Name (ID): Description
15:10	0h RO	**Reserved (RSVD):** Reserved
9	1h RW/O	**FLR Capability (FLRCAP):** A 1 in this bit indicates support for Function Level Reset (FLR).
8	1h RW/O	**TXP Capability (TXPCAP):** A 1 in this bit indicates support for the Transactions Pending (TXP) bit. TXP must be supported if FLR is supported.
7:0	6h RO	**Capability Length (CAPL):** This field indicates the # of bytes as required by the PCI spec. It has the value of 06h for the FLR Capability.

19.35 FLR Control (FLRCTL) — Offset B4h

This register shall be read-only 0 when SATAGC.FLRCSSEL=1.

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:18, F:0] + B4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:9	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
8	0h RO	**Transactions Pending (TXP):** A 1 indicates that the controller has issued Non-Posted request which has not been completed. A 0 indicates that Completions for all non-posted requests have been received by the controller.
7:1	0h RO	**Reserved (RSVD):** Reserved
0	0h RW	**Initiate FLR (INITIATE_FLR):** Used to initiated FLR transition. A write of 1 indicates FLR transition. Since hardware must not respond to any cycles till FLR completion, the value read by software from this bit is 0. Refer to Function Level Reset (FLR) on specifics to SATA.

19.36 Scratch Pad (SP) — Offset C0h

Scratch Pad

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + C0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Data (DT):** This is a read/write register that is available for software to use. No hardware action is taken on this register.

19.37 MSI-X Identifiers (MXID) — Offset D0h

MSI-X Identifiers

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:18, F:0] + D0h	11 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RW/L	**NEXT:** Indicates the next item in the list. This may be other capability pointers or it may be the last item in the list.The RW/L register attribute allows for flexibility in determining the capability structure available in this PCI function. Refer to SATAGC.REGLOCK description in order to lock the register to become RO. This register is not reset by FLR.
7:0	11h RO	**Capability ID (CID):** Capabilities ID indicates this is an MSI-X capability.

19.38 MSI-X Message Control (MXC) — Offset D2h

MSI-X Message Control

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:18, F:0] + D2h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**MSI-X Enable (MXE):** If set to '1' and the MSI Enable bit in the MSI Message Control register is cleared to '0', the function is permitted to use MSI-X to request service and is prohibited from using its INTx# pin (if implemented). If cleared to '0', the function is prohibited from using MSI-X to request service.
14	0h RW	**Function Mask (FM):** If set to '1', all of the vectors associated with the function are masked, regardless of their per vector Mask bit states. If cleared to '0', each vector's Mask bit determines whether the vector is masked or not. Setting or clearing the MSI-X Function Mask bit has no effect on the state of the per vector Mask bits.
13:11	0h RO	**Reserved (RSVD):** Reserved
10:0	0h RO	**Table Size (TS):** This value indicates the size of the MSI-X Table as the value n, which is encoded as n - 1. For example, a returned value of 3h corresponds to a table size of 4..

19.39 MSI-X Table Offset / Table BIR (MXT) — Offset D4h

MSI-X Table Offset / Table BIR

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + D4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RW/O	**Table Offset (TO):** Used as an offset from the address contained by one of the function's Base Address registers to point to the base of the MSI-X Table. The lower three Table BIR bits are masked off (cleared to 000b) by system software to form a 32-bit Qword-aligned offset.
2:0	0h RO	**Table BIR (TBIR):** This field indicates which one of a function's Base Address registers, located beginning at 10h in Configuration Space, is used to map the function's MSI-X Table into system memory. A read-only value of '0' means 10h.

19.40 MSI-X PBA Offset / PBA BIR (MXP) — Offset D8h

MSI-X PBA Offset / PBA BIR

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + D8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RW/O	**PBA Offset (PBAO):** Used as an offset from the address contained by one of the function's Base Address registers to point to the base of the MSI-X PBA. The lower three PBA BIR bits are masked off (cleared to 000b) by software to form a 32-bit Qword-aligned offset.
2:0	0h RO	**PBA BIR (PBIR):** This field indicates which one of a function's Base Address registers, located beginning at 10h in Configuration Space, is used to map the function's MSI-X PBA into system memory. A read-only value of '1' means 14h.

19.41 BIST FIS Control/Status (BFCS) — Offset E0h

BIST FIS Control/Status

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + E0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Reserved (RSVD):** Reserved
17	0h RW	**Port 7 BIST FIS Initiate (P7BFI):** When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 7, using the parameters specified in this register and BFTD1 and BFTD2. The BIST FIS will only be initiated if a device is present and not in the partial or slumber states. After a BIST FIS is successfully completed, software must disable and re-enable the PCS.P7E prior to attempting additional BIST FISes or to return the SATA controller to a normal operational mode. If the BIST FIS fails, as indicated by BFF in this register, software can clear and then set this bit to initiate another BIST FIS. This bit is only applicable to project(s): That has port 7 physically.
16	0h RW	**Port 6 BIST FIS Initiate (P6BFI):** When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 6, using the parameters specified in this register and BFTD1 and BFTD2. The BIST FIS will only be initiated if a device is present and not in the partial or slumber states. After a BIST FIS is successfully completed, software must disable and re-enable the PCS.P6E prior to attempting additional BIST FISes or to return the SATA controller to a normal operational mode. If the BIST FIS fails, as indicated by BFF in this register, software can clear and then set this bit to initiate another BIST FIS. This bit is only applicable to project(s): That has port 6 physically.
15	0h RW	**Port 5 BIST FIS Initiate (P5BFI):** When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 5, using the parameters specified in this register and BFTD1 and BFTD2. The BIST FIS will only be initiated if a device is present and not in the partial or slumber states. After a BIST FIS is successfully completed, software must disable and re-enable the PCS.P5E prior to attempting additional BIST FISes or to return the SATA controller to a normal operational mode. If the BIST FIS fails, as indicated by BFF in this register, software can clear and then set this bit to initiate another BIST FIS. This bit is only applicable to project(s): That has port 5 physically.
14	0h RW	**Port 4 BIST FIS Initiate (P4BFI):** When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 4, using the parameters specified in this register and BFTD1 and BFTD2. The BIST FIS will only be initiated if a device is present and not in the partial or slumber states. After a BIST FIS is successfully completed, software must disable and re-enable PCS.P4E prior to attempting additional BIST FISes or to return the SATA controller to a normal operational mode. If the BIST FIS fails, as indicated by BFF in this register, software can clear and then set this bit to initiate another BIST FIS. This bit is only applicable to project(s): That has port 4 physically.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**Port 3 BIST FIS Initiate (P3BFI):** When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 3, using the parameters specified in this register and BFTD1 and BFTD2. The BIST FIS will only be initiated if a device is present and not in the partial or slumber states. After a BIST FIS is successfully completed, software must disable and re-enable the PCS.P3E prior to attempting additional BIST FISes or to return the SATA controller to a normal operational mode. If the BIST FIS fails, as indicated by BFF in this register, software can clear and then set this bit to initiate another BIST FIS. This bit is only applicable to project(s): That has port 3 physically.
12	0h RW	**Port 2 BIST FIS Initiate (P2BFI):** When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 2, using the parameters specified in this register and BFTD1 and BFTD2. The BIST FIS will only be initiated if a device is present and not in the partial or slumber states. After a BIST FIS is successfully completed, software must disable and re-enable PCS.P2E prior to attempting additional BIST FISes or to return the SATA controller to a normal operational mode. If the BIST FIS fails, as indicated by BFF in this register, software can clear and then set this bit to initiate another BIST FIS. This bit is only applicable to project(s): That has port 2 physically.
11	0h RW/1C/V	**BIST FIS Successful (BFS):** This bit is set any time a BIST FIS transmitted by the SATA controller receives an R_OK completion status from the device.
10	0h RW/1C/V	**BIST FIS Failed (BFF):** This bit is set any time that a BIST FIS transmitted by the SATA controller receives an R_ERR completion status from the device.
9	0h RW	**Port 1 BIST FIS Initiate (P1BFI):** When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 1, using the parameters specified in this register and BFTD1 and BFTD2. The BIST FIS will only be initiated if a device is present and not in the partial or slumber states. After a BIST FIS is successfully completed, software must disable and re-enable PCS.P1E prior to attempting additional BIST FISes or to return the SATA controller to a normal operational mode. If the BIST FIS fails, as indicated by BFF in this register, software can clear and then set this bit to initiate another BIST FIS. This bit is only applicable to project(s): That has port 1 physically.
8	0h RW	**Port 0 BIST FIS Initiate (P0BFI):** When a rising edge is detected on this bit, the SATA controller will initiate a BIST FIS to the device on port 0, using the parameters specified in this register and BFTD1 and BFTD2. The BIST FIS will only be initiated if a device is present and not in the partial or slumber states. After a BIST FIS is successfully completed, software must disable and re-enable PCS.P0E prior to attempting additional BIST FISes or to return the SATA controller to a normal operational mode. If the BIST FIS fails, as indicated by BFF in this register, software can clear and then set this bit to initiate another BIST FIS. This bit is only applicable to project(s): That has port 0 physically.
7:2	0h RW	**BIST FIS Parameters (BFP):** These bits form the contents of the upper 6 bits of the BIST FIS Pattern Definition in the BIST FIS transmitted by the SATA controller. This field is not port specific - its contents will be used for any BIST FIS initiated on the SATA controller. The specific bit definitions are: Bit 7 (T) Far End Transmit mode; bit 6 (A) Align Bypass mode; bit 5 (S) Bypass Scrambling; bit 4 (L) Far End Retimed Loopback; bit 3 (F) Far End Analog Loopback; bit 2 (P) Primitive bit for use with Transmit mode.
1:0	0h RO	**Reserved (RSVD):** Reserved

19.42 BIST FIS Transmit Data 1 (BFTD1) — Offset E4h

BIST FIS Transmit Data 1

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + E4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DATA:** The data programmed into this register will form the contents of the second DW of any BIST FIS initiated by the SATA controller. This register is not port specific - its contents will be used for BIST FIS initiated on any port. Although the 2nd and 3rd DWs of the BIST FIS are only meaningful when the T bit of the BIST FIS is set to indicate Far-End Transmit mode, this register's contents will be transmitted as the BIST FIS 2nd DW regardless of whether or not the T bit is indicated in the BFCS register.

19.43 BIST FIS Transmit Data 2 (BFTD2) — Offset E8h

BIST FIS Transmit Data 2

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + E8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**DATA:** The data programmed into this register will form the contents of the third DW of any BIST FIS initiated by the SATA controller. This register is not port specific - its contents will be used for BIST FIS initiated on any port. Although the 2nd and 3rd DWs of the BIST FIS are only meaningful when the T bit of the BIST FIS is set to indicate Far-End Transmit mode, this register's contents will be transmitted as the BIST FIS 3rd DW regardless of whether or not the T bit is set in the BFCS register.

19.44 Manufacturing ID (MFID) — Offset F8h

Manufacturing ID

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:18, F:0] + F8h	8000FB1 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD):** Reserved
27:24	8h RO/V	**Dot portion of Process ID (DPID):** Indicates the dot. Process is reflected in bits [7:0]. This 8-bit value is received via the IOSF Sideband SetID message.
23:16	0h RO/V	**Stepping Identifier (SID):** This field is incremented for each stepping of the part. Note that this field can be used by software to differentiate steppings when the Revision ID may not change. This is SIP customer implementation specific and SIP customer provides this 8-bit value via the Sideband SetID message.
15:8	Fh RO/V	**Manufacturer Identifier (MID):** This is SIP customer implementation specific and SIP customer provides this 8-bit value via the Sideband SetID message.
7:0	B1h RO/V	**Process/Dot Identifier (PID):** Indicates the Process. Dot is reflected in bits [27:24]. This is SIP customer implementation specific and SIP customer provides this 8-bit value via the Sideband SetID message.

19.45 AHCI Index Register (INDEX)—Offset 10h

This registers are only available if CC.SCC is not 01h to index into all memory registers defined in Memory Registers and the message buffer used for enclosure management. If CC.SCC is 01h, these AHCI Index Data Pair registers are not accessible and SINDEX/SDATA register pair shall be used to index into a subset of the memory registers defined in (See Memory Registers for more information on which registers could be indexed).

Access Method

Type: IO Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	Reserved.
18:2	0h RW	**Index (INDEX):** This Index register is used to select the Dword offset of the Memory Mapped AHCI register to be accessed. A Dword, Word or Byte access is specified by the active byte enables of the I/O access to the Data register.
1:0	0h RO	Reserved.

19.46 AHCI Data Register (DATA)—Offset 14h

This registers are index into all memory registers defined in Memory Registers and the message buffer used for enclosure management.

Access Method

Type: IO Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Data (DATA):** This Data register is a window through which data is read or written to the memory mapped register pointed to by the Index register. Note that a physical register is not actually implemented as the data is actually stored in the memory mapped registers. Since this is not a physical register, the default value is the same as the default value of the register pointed to by Index.

19.47 HBA Capabilities (GHC_CAP)—Offset 0h

This register indicates basic capabilities of the HBA to driver software. The RWO bits in this register are only cleared upon PLTRST#. This register is not reset by FLR.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: FF36FF07h

Bit Range	Default & Access	Field Name (ID): Description
31	1h RW/O	**Supports 64-bit Addressing (S64A):** Indicates the S-ATA controller can access 64-bit data structures. The 32-bit upper bits of the port DMA Descriptor, the PRD Base, and each PRD entry are read/write.
30	1h RW/O	**Supports Native Command Queuing Acceleration (SCQA):** Indicates the SATA controller supports Serial-ATA Native Command Queueing. The HBA will handle DMA Setup FISes in hardware, including support for auto-activiate optimization through the FIS.
29	1h RW/O	**Supports SNotification Register (SSNTF):** When set to 1, indicates that the HBA supports the PxSNTF (SNotification) register and its associated functionality. When cleared to 0., the HBA does not support the PxSNTF (SNotification) register and its associated functionality.
28	1h RW/O	**Supports Mechanical Presence Switch (SMPS):** When set to 1, the HBA supports mechanical presence switches on its ports for use in hot plug operations. When cleared to 0, this function is not supported. This value is loaded by the BIOS prior to OS initialization..
27	1h RW/O	**Supports Staggered Spin-up (SSS):** Indicates whether the S-ATA controller supports staggered spin-up on its ports, for use in balancing power spikes. This value is loaded by platform BIOS prior to OS initiallization.
26	1h RW/O	**Supports Aggressive Link Power Management (SALP):** Indicates the S-ATA controller supports auto-generating link requests to the partial or slumber states when there are no commands to process. When cleared to 0, software shall treat the PxCMD.ALPE and PxCMD.ASP bits as reserved.
25	1h RW/O	**Supports Activity LED (SAL):** Indicates the S-ATA controller supports a single output pin (SATALED#) which indicates activity.
24	1h RW/O	**Supports Command List Override (SCLO):** When set to 1, indicates that the HBA supports the PxCMD.CLO bit and it's associated function. When cleared to 0., The HBA is not capable of clearing the BSY and DRQ bits in the Status register in order to issue a software reset if these bits are still set from a previous operation.

Bit Range	Default & Access	Field Name (ID): Description
23:20	3h RW/O	**Interface Speed Support (ISS):** Indicates the maximum speed the S-ATA controller can support on its ports. These encodings match the system software programmable PxSCTL.DET.SPD field. 0000 = Reserved; 0001 = Gen 1 (1.5 Gbps); 0010 = Gen 2 (3 Gbps); 0011 = Gen 3 (6 Gbps); 0100 - 1111 = Reserved. Note: if (FFSATA0p0, FFSATA0p1, FFSATA0p2, FFSATA0p3, FFSATA0p4 and FFSATA0p5) is 1, this field is RWO defaulting to 0010 and ignores software write value of 0011. If either FFSATA0p0, FFSATA0p1, , FFSATA0p2, FFSATA0p3, FFSATA0p4 or FFSATA0p5 is 0, this field is RWO defaulting to 0011.
19	0h RO	**Supports Non-Zero DMA Offsets (SNZO):** Reserved as per AHCI 1.3
18	1h RO	**Supports AHCI mode only (SAM):** The SATA controller may optionally support AHCI access mechanism only. A value of 0 indicates that in addition to the native AHCI mechanism (via ABAR), the SATA controller implements a legacy, task-file based register interface such as SFF-8038i. A value of 1 indicates that the SATA controller does not implement a legacy, task-file based register interface.
17	1h RO	**Supports Port Multiplier (SPM):** The SATA controller may optionally support command-based switching Port Multipliers. BIOS must clear this bit if Port Multipliers are not supported.
16	0h RO	**FIS-based Switching Supported (FBSS):** Not supported.
15	1h RO	**PIO Multiple DRQ Block (PMD):** If set to 1, the HBA supports multiple DRQ block data transfers for the PIO command protocol.
14	1h RW/O	**Slumber State Capable (SSC):** The SATA controller supports the slumber state.
13	1h RW/O	**Partial State Capable (PSC):** The SATA controller supports the partial state.
12:8	1Fh RO	**Number of Command Slots (NCS):** 1Fh indicating support for 32 slots.
7	0h RO	**Command Completion Coalescing Supported (CCCS):** When set to 1, indicates that the HBA supports command completion coalescing. When command completion coalescing is supported, the HBA has implemented the CCC_CTL and the CCC_PORTS global HBA registers. When cleared to 0, indicates that the HBA does not support command completion coalescing and the CCC_CTL and CCC_PORTS global HBA registers are not implemented.
6	0h RW/O/V	**Enclosure Management Supported (EMS):** When set to 1, indicates that the HBA supports enclosure management. When enclosure management is supported, the HBA has implemented the EM_LOC and EM_CTL global HBA registers. When cleared to 0, indicates that the HBA does not support enclosure management and the EM_LOC and EM_CTL global HBA registers are not implemented.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW/O	**Supports External SATA (SXS):** When set to 1, indicates that the HBA has one or more Serial ATA ports that has a signal only connector that is externally accessible. If this bit is set, software may refer to the PxCMD.ESP bit to determine whether a specific port has its signal connector externally accessible as a signal only connector (i.e. power is not part of that connector). When the bit is cleared to 0, indicates that the HBA has no Serial ATA ports that have a signal only connector externally accessible.
4:0	7h RO/V	**Number of Ports (NP):** 0's based value indicating the maximum number of ports supported. Note that the number of ports indicated in this field may be more than the number of ports indicated in the PI register. Number of ports shall be dependent on MAP.SC, fuses (FFSATA7, FFSATA8) and PCIe/SATA muxing configuration where if ANY of these parameter disable a particular port then that port is disabled and not counted. The maximum number of ports supported by SIP is 8 and the least is 0 (i.e. Function Disable). In the case of 0 port configuration, the value of NP is a don't care (while implementation has it fixed as 07h). Any combination in between is supported by SATA host controller. Indicates the number of supported ports.

19.48 Global HBA Control (GHC)—Offset 4h

This register controls various global actions of the HBA.

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 80000000h

Bit Range	Default & Access	Field Name (ID): Description
31	1h RO	**AHCI Enable (AE):** When set, indicates that an AHCI driver is loaded and communication to the HBA shall be via AHCI mechanisms. This can be used by an HBA that supports both legacy mechanisms (such as SFF-8038i) and AHCI to know when the HBA is running under an AHCI driver. When set, software shall only talk to the HBA using AHCI. The HBA will not have to allow command processing via both AHCI and legacy mechanisms. When cleared, software will only communicate with the HBA using legacy mechanisms. Software shall set this bit to 1 before accessing other AHCI registers. Note: The implementation of this bit is dependent upon the value of the CAP.SAM bit. If CAP.SAM is 0, then GHC.AE should be RW and shall have a reset value of 0. If CAP.SAM is 1, then AE shall be read only and shall have a reset value of 1.
30:3	0h RO	Reserved.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RO	**MSI Revert to Single Message (MRSM):** When set to 1 by hardware, indicates that the HBA requested more than one MSI vector but has reverted to using the first vector only. When this bit is cleared to 0, the HBA has not reverted to single MSI mode (i.e. hardware is already in single MSI mode, software has allocated the number of messages requested, or hardware is sharing interrupt vectors if MC.MME < MC.MMC). The HBA may revert to single MSI mode when the number of vectors allocated by the host is less than the number requested. This bit shall only be set to 1 when the following conditions hold: MC.MSIE = 1 (MSI is enabled); MC.MMC > 0 (multiple messages requested); MC.MME > 0 (more than one message allocated); MC.MME != MC.MMC (messages allocated not equal to number requested). When this bit is set to 1, single MSI mode operation is in use and software is responsible for clearing bits in the IS register to clear interrupts. This bit shall be cleared to 0 by hardware when any of the four conditions stated is false. This bit is also cleared to 0 when MC.MSIE = 1 and MC.MME = 0h. In this case, the hardware has been programmed to use single MSI mode, and is not reverting to that mode. The HBA shall always revert to single MSI mode when the number of vectors allocated by the host is leass than the number requested. Value of MRSM is a don't care when GHC.HR=1.
1	0h RW	**Interrupt Enable (IE):** This global bit enables interrupts from the HBA. When cleared (reset default), all interrupt sources from all ports are disabled. When set, interrupts are enabled.
0	0h RW/1S	**HBA Reset (HR):** When set by SW, this bit causes an internal reset of the HBA. All state machines that relate to data transfers and native command queuing will return to an idle condition, and all ports will be re-initialized via COMRESET. When the HBA has performed the reset action, it will reset this bit to 0. A software write of 0 will have no effect.

19.49 Interrupt Status Register (IS)—Offset 8h

This register indicates which of the ports within the controller have an interrupt pending and require service.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW/1C/V	**Interrupt Pending Status Port 7 (IPS7):** If set, indicates that port 7 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt. This bit is only applicable to project(s): That has port 7 physically.
6	0h RW/1C/V	**Interrupt Pending Status Port 6 (IPS6):** If set, indicates that port 6 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt. This bit is only applicable to project(s): That has port 6 physically.
5	0h RW/1C/V	**Interrupt Pending Status Port 5 (IPS5):** If set, indicates that port 5 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt. This bit is only applicable to project(s): That has port 5 physically.
4	0h RW/1C/V	**Interrupt Pending Status Port 4 (IPS4):** If set, indicates that port 4 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt. This bit is only applicable to project(s): That has port 4 physically.
3	0h RW/1C/V	**Interrupt Pending Status Port 3 (IPS3):** If set, indicates that port 3 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt. This bit is only applicable to project(s): That has port 3 physically.
2	0h RW/1C/V	**Interrupt Pending Status Port 2 (IPS2):** If set, indicates that port 2 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt. This bit is only applicable to project(s): That has port 2 physically.
1	0h RW/1C/V	**Interrupt Pending Status Port 1 (IPS1):** If set, indicates that port 1 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt. This bit is only applicable to project(s): That has port 1 physically.
0	0h RW/1C/V	**Interrupt Pending Status Port 0 (IPS0):** If set, indicates that port 0 has an interrupt pending. Software can use this information to determine which ports require service after an interrupt. This bit is only applicable to project(s): That has port 0 physically.

19.50 Ports Implemented (GHC_PI)—Offset Ch

This register indicates which ports are exposed to the HBA. It is loaded by platform BIOS. It indicates which ports that the device supports are available for software to use. Any available port may not be implemented. This register is not reset by FLR. There is BIOS programming requirement on the PI register.

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7	0h RW/O/V	**Port 7 Implemented (PI7):** If set, then port 7 is available for use. If cleared, the port is not available for use. This bit is reserved and is read-only 0 if below condition is true else it's RWO-zero: Refer to CAP.NP where port 7 is not available.
6	0h RW/O/V	**Port 6 Implemented (PI6):** If set, then port 6 is available for use. If cleared, the port is not available for use. This bit is reserved and is read-only 0 if below condition is true else it's RWO-zero: Refer to CAP.NP where port 6 is not available.
5	0h RW/O/V	**Port 5 Implemented (PI5):** If set, then port 5 is available for use. If cleared, the port is not available for use. This bit is reserved and is read-only 0 if below condition is true else it's RWO-zero: Refer to CAP.NP where port 5 is not available.
4	0h RW/O/V	**Port 4 Implemented (PI4):** If set, then port 4 is available for use. If cleared, the port is not available for use. This bit is reserved and is read-only 0 if below condition is true else it's RWO-zero: Refer to CAP.NP where port 4 is not available.
3	0h RW/O/V	**Port 3 Implemented (PI3):** If set, then port 3 is available for use. If cleared, the port is not available for use. This bit is reserved and is read-only 0 if below condition is true else it's RWO-zero: Refer to CAP.NP where port 3 is not available.
2	0h RW/O/V	**Port 2 Implemented (PI2):** If set, then port 2 is available for use. If cleared, the port is not available for use. This bit is reserved and is read-only 0 if below condition is true else it's RWO-zero: Refer to CAP.NP where port 2 is not available.
1	0h RW/O/V	**Port 1 Implemented (PI1):** If set, then port 1 is available for use. If cleared, the port is not available for use. This bit is reserved and is read-only 0 if below condition is true else it's RWO-zero: Refer to CAP.NP where port 1 is not available.
0	0h RW/O/V	**Port 0 Implemented (PI0):** If set, then port 0 is available for use. If cleared, the port is not available for use. This bit is reserved and is read-only 0 if below condition is true else it's RWO-zero: Refer to CAP.NP where port 0 is not available.

19.51 AHCI Version (VS)—Offset 10h

This register indicates the major and minor version of the AHCI specification. It is BCD encoded. The upper two bytes represent the major version number, and the lower two bytes represent the minor version number. Example: Version 3.12 would be represented as 00030102h.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 10300h

Bit Range	Default & Access	Field Name (ID): Description
31:16	1h RO	**Major Version Number (MJR):** Indicates the major version is 1.
15:0	300h RO	**Minor Version Number (MNR):** Indicates the minor version is 30.

19.52 Enclosure Management Location (EM_LOC)—Offset 1Ch

The enclosure management location register identifies the location and size of the enclosure management message buffer. This register is not implemented if enclosure management is not supported (i.e. CAP.EMS = 0).

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 1600002h

Bit Range	Default & Access	Field Name (ID): Description
31:16	160h RO	**Offset (OFST):** The offset of the message buffer in Dwords from the beginning of the ABAR.
15:0	2h RO	**Buffer Size (SZ):** Specifies the size of the transmit message buffer area in Dwords. If both transmit and receive buffers are supported, then the transmit buffer begins at ABAR[EM_LOC.OFST*4] and the receive buffer directly follows it. If both transmit and receive buffers are supported, both buffers are of the size indicated in the Buffer Size field. A value of 0 is invalid. Note that SATA controller only supports transmit buffer.

19.53 Enclosure Management Control (EM_CTL)—Offset 20h

This register is used to control and obtain status for the enclosure management interface. The register incluldes information on the attributes of the implementation, enclosure management messages supported, the status of the interface, whether any messages are pending, and is used to initiate sending messages. This register is not implemented if enclosure management is not supported (i.e. CAP.EMS = 0).

Access Metho.

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 7010000h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	Reserved.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**Port Multiplier Support (ATTR_PM):** The HBA does not support enclosure management messages for devices attached via a Port Multiplier. Software should use the Serial ATA enclosure management bridge that is built into many Port Multipliers for enclosure services with these devices. For more information on Serial ATA enclosure management bridges, refer to the Serial ATA II: Extensions to Serial ATA 1.0a revision 1.2 specification.
26	1h RW/O	**Activity LED Hardware Driven (ATTR_ALHD):** If set to 1, the HBA drives the activity LED for the LED message type in hardware and does not utilize software settings for this LED. The HBA does not begin transmitting the hardware based activity signal until after software has written CTL.TM=1 after a reset condition.
25	1h RO	**Transmit Only (ATTR_XMT):** If set to 1, the HBA only supports transmitting messages and does not support receiving messages. If cleared to 0, the HBA supports transmitting and receiving messages.
24	1h RO	**Single Message Buffer (ATTR_SMB):** If set to 1, the HBA has one message buffer that is shared for messages to transmit and messages received. In this case, unsolicited receive messages are not supported and it is software's responsibility to manage access to this buffer. If cleared to 0, there are separate receive and transmit buffers such that unsolicited messages could be supported.
23:20	0h RO	Reserved.
19	0h RO	**SGPIO Enclosure Management Messages (SUPP_SGPIO):** If set to 1, the HBA supports the SGPIO register interface message type.
18	0h RO	**SES-2 Enclosure Management Messages (SUPP_SES2):** If set to 1, the HBA supports the SES-2 message type.
17	0h RO	**SAF-TE Enclosure Management Messages (SUPP_SAFTE):** If set to 1, the HBA supports the SAF-TE message type.
16	1h RO	**LED Message Types (SUPP_LED):** If set to 1, the HBA supports the LED message type defined in LED Message Type.
15:10	0h RO	Reserved.
9	0h RW/1S	**Reset (RST):** When set to 1 by software, the HBA shall reset all enclosure management message logic and take all appropriate reset actions to ensure messages can be transmitted/received after the reset. After the HBA completes the reset operation, the HBA shall set the value to 0. A write of 0 by software to this field shall have no effect.

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW/1S	**Transmit Message (CTL_TM):** When set to 1 by software, the HBA shall transmit the message contained in the message buffer. When the message is completely sent, the HBA shall clear this bit to 0. A write of 0 to this bit by software shall have no effect. Software shall not change the contents of the message buffer while CTL.TM is set to 1.
7:1	0h RO	Reserved.
0	0h RO	**Message Received (STS_MR):** Message received is not supported.

19.54 HBA Capabilities Extended (GHC_CAP2)—Offset 24h

This register indicates basic capabilities of the HBA to driver software. The RWO bits in this register are only cleared upon PLTRST#. This register is not reset by FLR.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 3Ch

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RO	Reserved.
5	1h RW/O	**DEVSLP Entrance from Slumber Only (DESO):** This field specifies that the HBA shall only assert DEVSLP if the interface is in Slumber. When this bit is set to 1, the HBA shall ignore software directed entrance to DEVSLP via PxCMD.ICC unless PxSSTS.IPM = 6h. When this bit is cleared to 0, the HBA may enter DEVSLP from any link state (active, Partial, or Slumber). BIOS is required to program this field to 1.
4	1h RW/O/V	**Supports Aggressive DEVSLP Management (SADM):** When set to 1, the HBA supports hardware assertion of the DEVSLP signal after the idle timeout expires. When cleared to 0, this function is not supported and software shall treat the PxDEVSLP.ADSE field as reserved. Note: If PHY IO PM Disable Fuse is 1, this register will read 0. Else this register will read 1 with RWO attribute.
3	1h RW/O/V	**Supports DEVSLP (SDS):** When set to 1, the HBA supports the DEVSLP feature. When cleared to 0, DEVSLP is not supported. Note: If PHY IO PM Disable Fuse is 1, this register will read 0. Else this register will read 1 with RWO attribute.

Bit Range	Default & Access	Field Name (ID): Description
2	1h RW/O/V	**Automatic Partial to Slumber Transitions (APST):** When set to 1, the HBA supports Automatic Partial to Slumber Transitions. When cleared to 0, Automatic Partial to Slumber Transition is not supported. Note: If SATA PHY PM Disable Fuse is 1, this register will read only 0. Else this register will read 1 with RWO attribute.
1	0h RO	Reserved.
0	0h RO	**BIOS/OS Handoff (BOH):** Not supported.

19.55 Vendor Specific (VSP)—Offset A0h

Vendor Specific

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 48h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	Reserved.
6	1h RO	**Software Feature Mask Supported (SFMS):** Set to 1 if the platform is enabled for premium storage features mask (SFM) as described in VS MMIO space at offset ABAR[RPID.(OFST*4)+4].
5	0h RO/V	**Premium Features Supported (PFS):** Set to 1 if the platform is enabled for premium storage features (PFB) as described in VS MMIO space at offset ABAR[RPID.OFST*4]. Set to 0 if the platform is not enabled for premium storage features.
4	0h RO/V	**Platform Type (PT):** Set to 1 if mobile platform. Clear (0) if desktop.
3	1h RO	**Supports RAID Platform ID Reporting (SRPIR):** If set to 1, then indicates that the RAID Platform ID is reported via ABAR + C0h. Set to 0 if this ID is not reported via MMIO space.
2:0	0h RO	Reserved.

19.56 Vendor-Specific Capabilities Register (VS_CAP)—Offset A4h

Vendor-Specific Capabilities Register

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 1002DEh

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	Reserved.
27:16	10h RW/O	**NVM Remapped Register Offset (NRMO):** Specifies the offset (in 128B unit) within ABAR as to where the PCIe NAND memory BAR register space is remapped. For example, NRMO=1 means ABAR + 128B. This allows the remapped offset to shift between ABAR + 0B, and ABAR + 512KK - 128B, with 128B step. The remapped offset into the AHCI memory space must not overlap with the memory space used for SATA. The remapped offset into the AHCI memory space and the remapped size must not exceed the allocated AHCI memory space of the integrated AHCI controller. Only valid when NRMBE = 1. The reset default of this field is 10h, this places the start of the PCIe NAND memory BAR register space at ABAR + 2K. This field is not reset by FLR.
15:13	0h RO	Reserved.
12:1	16Fh RW/O	**Memory Space Limit. (MSL):** This field specifies the size (in 128B unit) of the remapped memory space for the PCIe NAND device. It is a 0-based field. For example, MSL=1 means 256B. This allows the remapped size to shift between 128B and 512K in steps of 128B. Memory BAR offset from 0 to MSL of the PCIe NAND device are remapped under the integrated AHCI controller memory space. The remapped offset into the AHCI memory space and the value programmed in this field must not exceed the allocated AHCI memory space of the integrated AHCI controller. Only valid when NRMBE = 1. The reset default of this field is 16Fh, which specifies the size of the remapped memory space as 34k. This field is not reset by FLR.
0	0h RW/O	**PCIe NAND Memory BAR Remapped Enable (NRMBE):** Set to 1 if a PCIe NAND device is present and remapping of its memory BAR register space is enabled. Cleared to 0 if there is no PCIe NAND device present or the remapping of its memory BAR register space is disabled. This bit is not reset by FLR.

19.57 Remapping Under NVMe (RUN)—Offset A8h

Remapping Under NVMe

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	Reserved.
16	0h RW/O	**Remapping Under NVMe Enable (RUNE):** When set to 1, AHCI Host Controller become a Remapped Device under the NVMe Host Controller hence the Bus Device Function of AHCI Host Controller is hidden from the OS. When this bit is 0, AHCI Host Controller is not remapped under NVMe Host Controller
15:8	0h RO	Reserved.
7:0	0h RW/O	**NVMe Device Function (NVMEDF):** Device and function number of NVMe Host Controller that AHCI Host Controller is remapped to. Bit 7:3 is the device number, bit 2:0 is the function number

19.58 RAID Platform ID (RPID)—Offset C0h

This register is used by the Intel Matrix Storage Manager OROM to match the features supported by the OROM with the platform on which the OROM is executing. This prevents the use of an OROM designed for newer chipsets from being use on older chipsets as this could reduce up-sell potential.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 311C02h

Bit Range	Default & Access	Field Name (ID): Description
31:16	31h RO	**Offset (OFST):** The offset of the Premium Feature Block (PFB) in DWords from the beginning of the ABAR. SFM follows directly after PFB.
15:0	1C02h RO/V	**RAID Platform ID (RPID):** Specifies the DID value that has been assigned to the platform. This is the same DID that is reported by the SATA controller when SATAGC.AIE is set to 1 except that the DID is always reported through this register, regardless if the programming of SATAGC.AIE.

19.59 Premium Feature Block (PFB)—Offset C4h

Note: Bits 4-0 are not bit-mapped to individual fuses and/or soft SKU settings; rather a single fuse FFSATA5& FFSATA 3 /soft sku is used to indicate support for all of these features (refer to VSP.PFS). These registers indicate to the Intel Rapid Storage Technology AHCI driver that those premium RAID features that can be supported on the platform.

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
15:5	0h RO	Reserved.
4	0h RO/V	**Reserved (RSVD_1):** Read value is the same as VSP.PFS.
3	0h RO/V	**Reserved (RSVD_2):** Read value is the same as VSP.PFS.
2	0h RO/V	**Reserved (RSVD_3):** Read value is the same as VSP.PFS.
1	0h RO/V	**Supports Email Alert (SEA):** Read value is the same as VSP.PFS. If set to 1, then this feature is supported as part of the premium feature set.
0	0h RO/V	**Supports OEM IOCTL (SOI):** Read value is the same as VSP.PFS. If set to 1, then this feature is supported as part of the premium feature set.

19.60 SW Feature Mask (SFM)—Offset C8h

The following will be programmed by the BIOS when VS_CAP.SFMS == 1. The feature mask is used by SW to determine which non-premium features shall be supported by SW. These register bits are not reset by FLR since they are programmed by BIOS.

Access Method

Type: MEM Register
(Size: 16 bits)

Device:
Function:

Default: 3Fh

Bit Range	Default & Access	Field Name (ID): Description
15:12	0h RO	Reserved.
11:10	0h RW/O	**OROM UI Normal Delay. (OROM_UI_Normal_Delay):** Values of these bits specify the delay of the OROM UI Splash Screen in a normal status. 00 = 2 secs (default and previous value); 01 = 4 secs; 10 = 6 secs; 11 = 8 secs. If bit 5 == 0, then these values are disregarded.
9	0h RW/O	**Smart Response Technology. (Smart_Response_Technology):** If set to '1', then Smart Response Technology is enabled. If cleared to '0', the feature is disabled.

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW/O	**RRT Only on ESATA (IRRT_Only_on_ESATA):** If set to 1, then only RRT volumes can span internal and external SATA ports (e.g. eSATA). If cleared to 0, then any RAID volume can span internal and external SATA ports (e.g. eSATA).
7	0h RW/O	**LED Locate (LED_Locate):** If set to 1, then LED/SGPIO hardware is attached and the ping to locate feature is enabled in the OS.
6	0h RW/O	**HDDUNLOCK (HDDUNLOCK):** If set to 1, then HDD password unlock is enabled in the OS.
5	1h RW/O	**OROM UI and BANNER (OROM_UI_and_BANNER):** If set to 1, then the OROM UI is displayed. When cleared to 0, the OROM UI and BANNER are not displayed if all disks and volumes have a normal status.
4	1h RW/O	**RRT (IRRT):** If set to 1, then Rapid Recovery Technology is enabled.
3	1h RW/O	**R5 (R5):** If set to 1, then RAID5 is enabled.
2	1h RW/O	**R10 (R10):** If set to 1, then RAID10 is enabled.
1	1h RW/O	**R1 (R1):** If set to 1, then RAID1 is enabled.
0	1h RW/O	**R0 (R0):** If set to 1, then RAID0 is enabled.

19.61 Port [0-7] Command List Base Address (PxCLB0)—Offset 100h

Port [0-7] Command List Base Address

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RW	**Command List Base Address (CLB):** Indicates the 32-bit base for the command list for this port. This base is used when fetching commands to execute. This address must be 1K aligned as indicated by bits 31:10 being read/write. Note that these bits are not reset on a HBA reset.
9:0	0h RO	Reserved.

19.62 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU0)—Offset 104h

Port [0-7] Command List Base Address Upper 32-bits

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Command List Base Address Upper (CLBU):** Indicates the upper 32-bits for the command list base address for this port. This base is used when fetching commands to execute. Note that these bits are not reset on a HBA reset.

19.63 Port [0-7] FIS Base Address (PxFB0)—Offset 108h

Port [0-7] FIS Base Address

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**FIS Base Address (FB):** Indicates the 32-bit base for received FISes. This address must be 256-byte aligned as indicated by bits 31:08 being read/write. When FIS-based switching is in use, this structure is 4KB in length and the address shall be 4KB aligned. Note that these bits are not reset on a HBA reset.
7:0	0h RO	Reserved.

19.64 Port [0-7] FIS Base Address Upper 32-bits (PxFBU0)—Offset 10Ch

Port [0-7] FIS Base Address Upper 32-bits

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**FIS Base Address Upper (FBU):** Indicates the upper 32-bits for the received FIS base for this port. Note that these bits are not reset on a HBA reset.

19.65 Port [0-7] Interrupt Status (PxIS0)—Offset 110h

Port [0-7] Interrupt Status

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Cold Presence Detect Status (CPDS):** The SATA controller does not support cold presence detect.
30	0h RW/1C/V	**Task File Error Status (TFES):** This bit is set whenever the status register is updated by the device and the error bit (bit 0 of the Status field in the received FIS) is set.
29	0h RW/1C/V	**Host Bus Fatal Error Status (HBFS):** Indicates that the HBA encountered a host bus error that it cannot recover from, such as a bad software pointer. In PCI, such an indication would be a target or master abort.
28	0h RW/1C/V	**Host Bus Data Error Status (HBDS):** Indicates that the HBA encountered a data error (uncorrectable ECC / parity) when reading from or writing to system memory.
27	0h RW/1C/V	**Interface Fatal Error Status (IFS):** Indicates that the HBA encountered an error on the SATA interface which caused the transfer to stop.
26	0h RW/1C/V	**Interface Non-fatal Error Status (INFS):** Indicates that the HBA encountered an error on the SATA interface but was able to continue operation.
25	0h RO	Reserved.
24	0h RW/1C/V	**Overflow Status (OFS):** Indicates that the HBA received more bytes from a device than was specified in the PRD table for the command.
23	0h RW/1C/V	**Incorrect Port Multiplier Status (IPMS):** Indicates that the HBA received a FIS from a device whose port multiplier field did not match what was expected. The IPMS bit may be set during emuneration process. It is recommended that IPMS only be used after enumeration is complete on the Port Multiplier.

Bit Range	Default & Access	Field Name (ID): Description
22	0h RO/V	**PhyRdy Change Status (PRCS):** When set to one indicates the internal PhyRdy signal changed state. This bit reflects the state of PxSERR.DIAG.N. This bit is RO and is only cleared when PxSERR.DIAG.N is cleared.
21:8	0h RO	Reserved.
7	0h RW/1C/V	**Device Mechanical Presence Status (DMPS):** When set, indicates that a platform mechanical presence switch has been opened or closed, which may lead to a change in the connection state of the device. This bit is only valid in systems that support mechanical presence switch (CAP.SMPS and PxCMD.MPSP are set).
6	0h RO/V	**Port Connect Change Status (PCS):** • 1: Change in Current Connect Status. • 0: No change in Current Connect Status. This bit reflects the state of PxSERR.DIAG.X. This bit is only cleared when PxSERR.DIAG.X is cleared.
5	0h RW/1C/V	**Descriptor Processed (DPS):** A PRD with the I. bit set has transferred all of its data.
4	0h RO/V	**Unknown FIS Interrupt (UFS):** When set to 1 indicates that an unknown FIS was received and has been copied into system memory. This bit is cleared to 0 by software clearing the PxSERR.DIAG.F bit to 0. Note that this bit does not directly reflect the PxSERR.DIAG.F bit. PxSERR.DIAG.F is set immediately when an unknown FIS is detected, whereas this bit is set when the FIS is posted to memory. Software should wait to act on an unknown FIS until this bit is set to 1 or the two bits may become out of sync.
3	0h RW/1C/V	**Set Device Bits Interrupt (SDBS):** A Set Device Bits FIS has been received with the I. bit set and has been copied into system memory.
2	0h RW/1C/V	**DMA Setup FIS Interrupt (DSS):** A DMA Setup FIS has been received with the I. bit set and has been copied into system memory.
1	0h RW/1C/V	**PIO Setup FIS Interrupt (PSS):** A PIO Setup FIS has been received with the I. bit set, it has been copied into system memory, and the data related to that FIS has been transferred. This bit shall be set even if the data transfer resulted in an error.
0	0h RW/1C/V	**Device to Host Register FIS Interrupt (DHRS):** A D2H register FIS has been received with the I. bit set, and has been copied into system memory.

19.66 Port [0-7] Interrupt Enable (PxIE0)—Offset 114h

Port [0-7] Interrupt Enable

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Cold Presence Detect Enable (CPDS):** The SATA controller does not support cold presence detect.
30	0h RW	**Task File Error Enable (TFEE):** When set, GHC.IE is set, and P0S.TFES is set, the HBA shall generate an interrupt.
29	0h RW	**Host Bus Fatal Error Enable (HBFE):** When set, GHC.IE is set, and P0IS.HBFS is set, the HBA shall generate an interrupt.
28	0h RW	**Host Bus Data Error Enable (HBDE):** when set, GHC.IE is set, and P0IS.HBDS is set, the HBA shall generate an interrupt.
27	0h RW	**Interface Fatal Error Enable (IFE):** When set, GHC.IE is set, and P0IS.IFS is set, the HBA shall generate an interrupt.
26	0h RW	**Interface Non-fatal Error Enable (INFE):** When set, GHC.IE is set, and P0IS.INFS is set, the HBA shall generate an interrupt.
25	0h RO	Reserved.
24	0h RW	**Overflow Enable (OFE):** When set, and GHC.IE and P0IS.OFS are set, the HBA shall generate an interrupt.
23	0h RW	**Incorrect Port Multiplier Enable (IPME):** When set, and GHC.IE and P0IS.IPMS are set, the HBA shall generate an interrupt. BIOS is required to program this field to '0'. The same applies to AHCI driver.
22	0h RW	**PhyRdy Change Interrupt Enable (PRCE):** When set, and GHC.IE is set, and PxIS.PRCS is set, the HBA shall generate an interrupt.
21:8	0h RO	Reserved.
7	0h RW	**Device Mechanical Enable (DMPE):** When set, and P0IS.DMPS is set, the HBA shall generate an interrupt.
6	0h RW/V	**Port Change Interrupt Enable (PCE):** When set, GHC.IE is set, and P0IS.PCS is set, the HBA shall generate an interrupt.
5	0h RW	**Descriptor Processed Interrupt Enable (DPE):** When set, GHC.IE is set, and P0IS.DPS is set, the HBA shall generate an interrupt.
4	0h RW	**Unknown FIS Interrupt Enable (UFE):** When set, GHC.IE is set, and PxIS.UFS is set to 1, the HBA shall generate an interrupt.
3	0h RW	**Set Device Bits FIS Interrupt Enable (SDBE):** When set, GHC.IE is set, and P0IS.SDBS is set, the HBA shall generate an interrupt.
2	0h RW	**DMA Setup FIS Interrupt Enable (DSE):** When set, GHC.IE is set, and P0IS.DSS is set, the HBA shall generate an interrupt.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**PIO Setup FIS Interrupt Enable (PSE):** When set, GHC.IE is set, and P0IS.PSS is set, the HBA shall generate an interrupt.
0	0h RW	**Device to Host Register FIS Interrupt Enable (DHRE):** When set, GHC.IE is set, and P0IS.DHRS is set, the HBA shall generate an interrupt.

19.67 Port [0-7] Command (PxCMD0)—Offset 118h

Port [0-7] Command

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 4h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Interface Communication Control (ICC):** This is a four-bit field which can be used to control reset and power states of the interface. If the Link layer is currently in the L_IDLE state, writes to this field shall cause the HBA to initiate a transition to the interface power management state requested. If the Link layer is not currently in the L_IDLE state, writes to this field shall have no effect.
27	0h RW	**Aggressive Slumber Partial (ASP):** When set to 1, and the ALPE bit is set, the HBA shall aggressively enter the Slumber state when it clears a bit in the PxCI or PxSACT register and the register values are then PxCI = 0h and PxSACT = 0h. When cleared, and the ALPE bit is set, the HBA will aggressively enter the Partial state when it clears a bit in the PxCI or PxSACT register and the register values are then PxCI = 0h and PxSACT = 0h. If CAP.SALP is cleard to 0., software shall treat this bit as reserved.
26	0h RW	**Aggressive Link Power Management Enable (ALPE):** When set, the HBA will aggressively enter a lower link power state (Partial or Slumber) based upon the setting of the ASP bit. Software shall only set this bit to 1 if CAP.SALP is set to 1. If CAP.SALP is cleared to 0, software shall treat this bit as reserved. BIOS is recommeded to program this field to 1.
25	0h RW	**Drive LED on ATAPI Enable (DLAE):** When set, the HBA will drive the LED pin active for commands regardless of the state of PxCMD.ATAPI. When cleared, the HBA will only drive the LED pin active for commands if PxCMD.ATAPI is set to 0.. This bit is set by software
24	0h RW	**Device is ATAPI (ATAPI):** When set, the connected device is an ATAPI device. This bit is used by the HBA to control whether or not to generate the desktop LED when commands are active.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**Automatic Partial to Slumber Transitions Enabled (APSTE):** When set to 1., the HBA may perform Automatic Partial to Slumber Transitions. When cleared to 0. the port shall not perform Automatic Partial to Slumber Transitions. Software shall only set this bit to 1. if CAP2.APST is set to 1.; if CAP2.APST is cleared to 0. software shall treat this bit as reserved.
22	0h RO	**FIS-based Switching Capable Port (FBSCP):** The SATA controller does not support FIS-Based Switching.
21	0h RW/O	**External SATA Port (ESP):** When set to 1, indicates that this port is routed externally and will be used with an external SATA device. When set to 1, CAP.SXS must also be set to 1. When cleared (0), indicates that this port is not routed externally and supports internal SATA devices only. If ESP is set to '1', then the port may experience hot plug events.
20	0h RO	**Cold Presence Detection (CPD):** The SATA controller does not support cold presence detect.
19	0h RW/O	**Mechanical Presence Switch Attached to Port (MPSP):** If set to 1, the platform supports a mechanical presence switch attached to this port. If cleared to 0, the platform does not support a mechanical presence switch attached to this port.
18	0h RW/O	**Hot Plug Capable Port (HPCP):** This indicates whether the this port is connected to a device which can be hot plugged. SATA by definition is hot-pluggable, but not all platforms are constructed to allow the device to be removed (it may be screwed into the chassis, for example). This bit can be used by system software to indicate a feature such as eject device to the end-user.
17	0h RW/V	**Port Multiplier Attached (PMA):** When set, a Port Multiplier is attached to the HBA for this port. When cleared, a Port Multiplier is not attached to the HBA for this port. This bit is a read only 0. when CAP.PMS = 0., and read/write when CAP.PMS = 1.. Note that this bit is set by software; hardware does not auto-detect that a Port Multiplier is attached.
16	0h RO	Reserved.
15	0h RO/V	**Command List Running (CR):** When this bit is set it indicates that the command list DMA engine for the port is running.
14	0h RO/V	**FIS Receive Running (FR):** When this bit is set it indicates that the FIS Receive DMA engine for the port is running. Note to software: When FR bit stays set, please read the PxIS.PCS and PxSERR.DET.X to service the PCS interrupt accordingly if any
13	0h RO/V	**Mechanical Presence Switch State (MPSS):** The MPSS bit reports the state of a mechanical presence switch attached to this port. If CAP.SMPS is set to 1 and the mechanical presence switch is closed then this bit is cleared to 0. If CAP.SMPS is set to 1 and the mechanical presence switch is open then this bit is set to 1. If CAP.SMPS is set to 0 then this bit is cleared to 0. Software should only use this bit if both CAP.SMPS and PxCMD.MPSP are set to 1.

Bit Range	Default & Access	Field Name (ID): Description
12:8	0h RO/V	**Current Command Slot (CCS):** Indicates the current command slot the HBA is processing. This field is valid when the PxCMD.ST bit is set, and is constantly updated by the HBA. This field can be updated as soon as the HBA recognizes an active command slot, or at some point soon after when it begins processing the command. When PxCMD.ST transitions from a 1 to a 0, the HBA will reset this field to 0. After PxCMD.ST transitions from 0 to 1, the highest priority slot to issue from next is command slot 0. After the first command has been issued, the highest priority slot to issue from next is PxCMD.CCS + 1. For example, after the HBA has issued its first command, if PxCMD.CCS = 0h and PxCI is set to 3h, the next command that will be issued is from command slot 1.
7:5	0h RO	Reserved.
4	0h RW	**FIS Receive Enable (FRE):** When set, the HBA may post received FISes into the FIS receive area pointed to by PxFB and PxFBU. When cleared, received FISes are not accepted by the HBA, except for the first D2H register FIS after the initialization sequence.
3	0h RW/1S	**Command List Override (CLO):** Setting this bit to 1 causes PxTFD.STS.BSY and PxTFD.STS.DRQ to be cleared to 0. This allows a software reset to be transmitted to the device regardless of whether the BSY and DRQ bits are still set in the PxTFD.STS register. The HBA sets this bit to 0 when PxTFD.STS.BSY and PxTFD.STS.DRQ have been cleared to 0. A write to this register with a value of 0 shall have no effect.
2	1h RO	**Power On Device (POD):** The SATA controller does not support cold presence detect.
1	0h RW/V	**Spin-Up Device (SUD):** This bit is read/write for HBAs that support staggered spin-up via CAP.SSS. This bit is read only 1. for HBAs that do not support staggered spin-up. On an edge detect from 0. to 1, the HBA shall start a COMRESET initializatoin sequence to the device. Clearing this bit causes no action on the interface. Clearing this bit to0 does not cause any OOB signal to be sent on the interface. When this bit is cleared to 0. and PxSCTL.DET = 0h, the HBA will enter listen mode.
0	0h RW	**Start (ST):** When set, the HBA may process the command list. When cleared, the HBA may not process the command list. Whenever this bit is changed from a 0 to a 1, the HBA starts processing the command list at entry 0. Whenever this bit is changed from a 1 ot a 0, the PxCI and PxSACT register is cleared by the HBA upon the HBA putting the controller into an idle state. PxTFD shall be updated also.

19.68 Port [0-7] Task File Data (PxTFD0)—Offset 120h

Port [0-7] Task File Data

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 7Fh

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	Reserved.
15:8	0h RO/V	**Error (ERR):** Contains the latest copy of the task file error register.
7	0h RO/V	**Status Busy (STS_BSY):** Status - Indicates the interface is busy.
6:4	7h RO/V	**Status Rsvd0 (STS_RSVD0):** Status - Not Applicable.
3	1h RO/V	**Status Drq (STS_DRQ):** Status - Indicates a data transfer is requested.
2:1	3h RO/V	**Status Rsvd1 (STS_RSVD1):** Status - Not Applicable.
0	1h RO/V	**Status Err (STS_ERR):** Status - Indicates an error during the transfer.

19.69 Port [0-7] Signature (PxSIG0)—Offset 124h

Port [0-7] Signature

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: FFFFFFFFh

Bit Range	Default & Access	Field Name (ID): Description
31:0	FFFFFFFFh RO/V	**Signature (SIG):** Contains the signature received from a device on the first D2H Register FIS.

19.70 Port [0-7] Serial ATA Status (PxSSTS0)—Offset 128h

Port [0-7] Serial ATA Status

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	Reserved.
11:8	0h RO/V	**Interface Power Management (IPM):** Indicates the current interface state
7:4	0h RO/V	**Current Interface Speed (SPD):** Indicates the negotiated interface communication speed.
3:0	0h RO/V	**Device Detection (DET):** Indicates the interface device detection and Phy state.

19.71 Port [0-7] Serial ATA Control (PxSCTL0)—Offset 12Ch

Port [0-7] Serial ATA Control

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RO	Reserved.
19:16	0h RW	**Port Multiplier Port (PMP):** This field is not used by AHCI.
15:12	0h RW	**Select Power Management (SPM):** This field is not used by AHCI.
11:8	0h RW	**Interface Power Management Transitions Allowed (IPM):** Indicates which power states the HBA is allowed to transition to. If an interface power management state is not allowed via this register field, the HBA will not initiate that state and the HBA will PMNAKP any request from the device to enter that state.
7:4	0h RW	**Speed Allowed (SPD):** Indicates the highest allowable speed of the interface.
3:0	0h RW	**Device Detection Initialization (DET):** Controls the HBA.s device detection and interface initialization.

19.72 Port [0-7] Serial ATA Error (PxSERR0)—Offset 130h

Port [0-7] Serial ATA Error

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/1C/V	**Diagnostics (DIAG):** Contains diagnostic error information for use by diagnostic software in validating correct operation or isolating failure modes.
15:0	0h RW/1C/V	**Error (ERR):** The ERR field contains error information for use by host software in determining the appropriate response to the error condition. If one or more of bits 11:8 of this register are set, the controller will stop the current transfer.

19.73 Port [0-7] Serial ATA Active (PxSACT0)—Offset 134h

Port [0-7] Serial ATA Active

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/1S/V	**Device Status (DS):** System software sets this bit for native command queuing commands prior to setting the PxCI.CI bit in the same command slot entry. This field is cleared via the Set Device Bits FIS.

19.74 Port [0-7] Commands Issued (PxCI0)—Offset 138h

Port [0-7] Commands Issued

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/1S/V	**Commands Issued (CI):** This field is set by software to indicate to the HBA that a command has been built in system memory for a command slot and may be sent to the device. When the HBA receives a FIS which clears the BSY, ERR, and DRQ bits for the command, it clears the corresponding bit in this register for that command slot. Bits in this field shall only be set to 1 by software when PxCMD.ST is set to 1.

19.75 Port [0-7] SNotification (PxSNTF0)—Offset 13Ch

Port [0-7] SNotification

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	Reserved.
15:0	0h RW/1C/V	**PM Notify (PMN):** This field indicates whether a particular device with the corresponding PM Port number issued a Set Device Bits FIS to the host with the Notification bit set. PM Port 0h sets bit 0. PM Port Fh sets bit 15. Individual bits are cleared by software writing 1's to the corresponding bit positions. Note that, while this field is reset to default on a HBA Reset, it is not reset by COMRESET or SRST

19.76 Port [0-7] Device Sleep (PxDEVSLP0)—Offset 144h

Port [0-7] Device Sleep

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 1E022852h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	Reserved.
28:25	Fh RW/V	**DITO Multiplier (DM):** 0's based value that specifies the DITO multiplier that the HBA applies to the specified DITO value, effectively extending the range of DITO from 1ms to 16368ms. A value of 0h indicates a multiplier of 1. A maximum multiplier of 16 may be applied. The HBA computes the total idle timeout as a product of DM and DITO (i.e. DITO actual = DITO * DM).

Bit Range	Default & Access	Field Name (ID): Description
24:15	4h RW/V	**DEVSLP Idle Timeout (DITO):** This field specifies the amount of the time (in approximate 1ms granularity) that the HBA shall wait before driving the DEVSLP signal.Hardware reloads its port specific DEVSLP timer with this value each time the port transitions out of DEVSLP state. For example: from DEVSLP to active or PxDEVSLP.ADSE transitions from 0 to a 1. If CAP2.SDS or CAP2.SADM or PxDEVSLP.DSP is cleared to 0 then these bits are read-only 0h and software shall treat these bits as reserved. Software shall only set this value when PxCMD.ST is cleared to 0 and PxDEVSLP.ADSE is cleared to 0.
14:10	Ah RW/V	**DEVSLP Minimum Assertion Time (MDAT):** This field specifies the minimum amount of time (in 1ms granularity) that the HBA must assert the DEVSLP signal before it may be de-asserted. The nominal value is 10ms and the minimum is 1ms depending on device identification information. If CAP2.SDS is cleared to 0 or PxDEVSLP.DSP is cleared to 0 then these bits are read-only 0h and software shall treat these bits as reserved. Software shall only set this value when PxCMD.ST is cleared to 0, PxDEVSLP.ADSE is cleared to 0, and prior to setting PxCMD.ICC to 8h.
9:2	14h RW/V	**DEVSLP Exit Timeout (DETO):** This field specifies the maximum duration (in approximate 1ms granularity) from DEVSLP de-assertion until the device is ready to accept OOB. The nominal value is 20ms while the max value is 255ms depending on device identification information. If CAP2.SDS is cleared to 0 or PxDEVSLP.DSP is cleared to 0 then these bits are read-only 0h and software shall treat these bits as reserved. Software shall only set this value when PxCMD.ST is cleared to 0, PxDEVSLP.ADSE is cleared to 0, and prior to setting PxCMD.ICC to 8h.
1	1h RW/O	**DEVSLP Present (DSP):** If set to '1', the platform supports DEVSLP on this port. If cleared to '0', the platform does not support DEVSLP on this port. This bit may only be set to '1' if CAP2.SDS is set to '1'.DSP is mutually exclusive with the PxCMD.HPCP bit and PxCMD.ESP bit. Note that these bits are not reset on a HBA reset.
0	0h RW/V	**Aggressive DEVSLP Enable (ADSE):** This bit is read/write for HBAs that support aggressive DEVSLP management (CAP2. SADM = 1). When this bit is set to 1, the HBA shall assert the DEVSLP signal after the port has been idle (PxCI = 0h and PxSACT = 0h) for the amount of time specified by the PxDEVSLP.DITO register and the interface is in Slumber (PxSSTS.IPM = 6h). When this bit is cleared to 0, the HBA does not enter DEVSLP unless software directed via PxCMD.ICC. This bit shall only be set to 1 if PxDEVSLP.DSP is set to 1. If this bit is set to 1 and software clears the bit to 0, then the HBA shall de-assert the DEVSLP signal if asserted. Note that these bits are not reset on a HBA reset. BIOS is recommended to program this field to 1 if the platform support the DEVSLP feature. If CAP2.SDS is cleared to 0 or CAP2.SADM is cleared to 0, or if PxDEVSLP.DSP is cleared to 0 then these bits are read-only 0h and software shall treat these bits as reserved.

19.77 Port [0-7] Command List Base Address (PxCLB1)—Offset 180h

Port [0-7] Command List Base Address

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RW	**Command List Base Address (CLB):** Indicates the 32-bit base for the command list for this port. This base is used when fetching commands to execute. This address must be 1K aligned as indicated by bits 31:10 being read/write. Note that these bits are not reset on a HBA reset.
9:0	0h RO	Reserved.

19.78 Port [0-7] Command List Base Address Upper 32-bits (PxCLBU1)—Offset 184h

Port [0-7] Command List Base Address Upper 32-bits

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Command List Base Address Upper (CLBU):** Indicates the upper 32-bits for the command list base address for this port. This base is used when fetching commands to execute. Note that these bits are not reset on a HBA reset.

19.79 Port [0-7] FIS Base Address (PxFB1)—Offset 188h

Port [0-7] FIS Base Address

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**FIS Base Address (FB):** Indicates the 32-bit base for received FISes. This address must be 256-byte aligned as indicated by bits 31:08 being read/write. When FIS-based switching is in use, this structure is 4KB in length and the address shall be 4KB aligned. Note that these bits are not reset on a HBA reset.
7:0	0h RO	Reserved.

19.80 Port [0-7] FIS Base Address Upper 32-bits (PxFBU1)—Offset 18Ch

Port [0-7] FIS Base Address Upper 32-bits

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**FIS Base Address Upper (FBU):** Indicates the upper 32-bits for the received FIS base for this port. Note that these bits are not reset on a HBA reset.

19.81 Port [0-7] Interrupt Status (PxIS1)—Offset 190h

Port [0-7] Interrupt Status

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Cold Presence Detect Status (CPDS):** The SATA controller does not support cold presence detect.
30	0h RW/1C/V	**Task File Error Status (TFES):** This bit is set whenever the status register is updated by the device and the error bit (bit 0 of the Status field in the received FIS) is set.
29	0h RW/1C/V	**Host Bus Fatal Error Status (HBFS):** Indicates that the HBA encountered a host bus error that it cannot recover from, such as a bad software pointer. In PCI, such an indication would be a target or master abort.

Bit Range	Default & Access	Field Name (ID): Description
28	0h RW/1C/V	**Host Bus Data Error Status (HBDS):** Indicates that the HBA encountered a data error (uncorrectable ECC / parity) when reading from or writing to system memory.
27	0h RW/1C/V	**Interface Fatal Error Status (IFS):** Indicates that the HBA encountered an error on the SATA interface which caused the transfer to stop.
26	0h RW/1C/V	**Interface Non-fatal Error Status (INFS):** Indicates that the HBA encountered an error on the SATA interface but was able to continue operation.
25	0h RO	Reserved.
24	0h RW/1C/V	**Overflow Status (OFS):** Indicates that the HBA received more bytes from a device than was specified in the PRD table for the command.
23	0h RW/1C/V	**Incorrect Port Multiplier Status (IPMS):** Indicates that the HBA received a FIS from a device whose port multiplier field did not match what was expected. The IPMS bit may be set during emuneration process. It is recommended that IPMS only be used after enumeration is complete on the Port Multiplier.
22	0h RO/V	**PhyRdy Change Status (PRCS):** When set to one indicates the internal PhyRdy signal changed state. This bit reflects the state of PxSERR.DIAG.N. This bit is RO and is only cleared when PxSERR.DIAG.N is cleared.
21:8	0h RO	Reserved.
7	0h RW/1C/V	**Device Mechanical Presence Status (DMPS):** When set, indicates that a platform mechanical presence switch has been opened or closed, which may lead to a change in the connection state of the device. This bit is only valid in systems that support mechanical presence switch (CAP.SMPS and PxCMD.MPSP are set).
6	0h RO/V	**Port Connect Change Status (PCS):** • 1: Change in Current Connect Status. • 0: No change in Current Connect Status. This bit reflects the state of PxSERR.DIAG.X. This bit is only cleared when PxSERR.DIAG.X is cleared.
5	0h RW/1C/V	**Descriptor Processed (DPS):** A PRD with the I. bit set has transferred all of its data.
4	0h RO/V	**Unknown FIS Interrupt (UFS):** When set to 1 indicates that an unknown FIS was received and has been copied into system memory. This bit is cleared to 0 by software clearing the PxSERR.DIAG.F bit to 0. Note that this bit does not directly reflect the PxSERR.DIAG.F bit. PxSERR.DIAG.F is set immediately when an unknown FIS is detected, whereas this bit is set when the FIS is posted to memory. Software should wait to act on an unknown FIS until this bit is set to 1 or the two bits may become out of sync.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW/1C/V	**Set Device Bits Interrupt (SDBS):** A Set Device Bits FIS has been received with the I. bit set and has been copied into system memory.
2	0h RW/1C/V	**DMA Setup FIS Interrupt (DSS):** A DMA Setup FIS has been received with the I. bit set and has been copied into system memory.
1	0h RW/1C/V	**PIO Setup FIS Interrupt (PSS):** A PIO Setup FIS has been received with the I. bit set, it has been copied into system memory, and the data related to that FIS has been transferred. This bit shall be set even if the data transfer resulted in an error.
0	0h RW/1C/V	**Device to Host Register FIS Interrupt (DHRS):** A D2H register FIS has been received with the I. bit set, and has been copied into system memory.

19.82 Port [0-7] Interrupt Enable (PxIE1)—Offset 194h

Port [0-7] Interrupt Enable

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Cold Presence Detect Enable (CPDS):** The SATA controller does not support cold presence detect.
30	0h RW	**Task File Error Enable (TFEE):** When set, GHC.IE is set, and P0S.TFES is set, the HBA shall generate an interrupt.
29	0h RW	**Host Bus Fatal Error Enable (HBFE):** When set, GHC.IE is set, and P0IS.HBFS is set, the HBA shall generate an interrupt.
28	0h RW	**Host Bus Data Error Enable (HBDE):** when set, GHC.IE is set, and P0IS.HBDS is set, the HBA shall generate an interrupt.
27	0h RW	**Interface Fatal Error Enable (IFE):** When set, GHC.IE is set, and P0IS.IFS is set, the HBA shall generate an interrupt.
26	0h RW	**Interface Non-fatal Error Enable (INFE):** When set, GHC.IE is set, and P0IS.INFS is set, the HBA shall generate an interrupt.
25	0h RO	Reserved.
24	0h RW	**Overflow Enable (OFE):** When set, and GHC.IE and P0IS.OFS are set, the HBA shall generate an interrupt.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**Incorrect Port Multiplier Enable (IPME):** When set, and GHC.IE and P0IS.IPMS are set, the HBA shall generate an interrupt. BIOS is required to program this field to '0'. The same applies to AHCI driver.
22	0h RW	**PhyRdy Change Interrupt Enable (PRCE):** When set, and GHC.IE is set, and PxIS.PRCS is set, the HBA shall generate an interrupt.
21:8	0h RO	Reserved.
7	0h RW	**Device Mechanical Enable (DMPE):** When set, and P0IS.DMPS is set, the HBA shall generate an interrupt.
6	0h RW/V	**Port Change Interrupt Enable (PCE):** When set, GHC.IE is set, and P0IS.PCS is set, the HBA shall generate an interrupt.
5	0h RW	**Descriptor Processed Interrupt Enable (DPE):** When set, GHC.IE is set, and P0IS.DPS is set, the HBA shall generate an interrupt.
4	0h RW	**Unknown FIS Interrupt Enable (UFE):** When set, GHC.IE is set, and PxIS.UFS is set to 1, the HBA shall generate an interrupt.
3	0h RW	**Set Device Bits FIS Interrupt Enable (SDBE):** When set, GHC.IE is set, and P0IS.SDBS is set, the HBA shall generate an interrupt.
2	0h RW	**DMA Setup FIS Interrupt Enable (DSE):** When set, GHC.IE is set, and P0IS.DSS is set, the HBA shall generate an interrupt.
1	0h RW	**PIO Setup FIS Interrupt Enable (PSE):** When set, GHC.IE is set, and P0IS.PSS is set, the HBA shall generate an interrupt.
0	0h RW	**Device to Host Register FIS Interrupt Enable (DHRE):** When set, GHC.IE is set, and P0IS.DHRS is set, the HBA shall generate an interrupt.

19.83 Port [0-7] Command (PxCMD1)—Offset 198h

Port [0-7] Command

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 4h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**Interface Communication Control (ICC):** This is a four-bit field which can be used to control reset and power states of the interface. If the Link layer is currently in the L_IDLE state, writes to this field shall cause the HBA to initiate a transition to the interface power management state requested. If the Link layer is not currently in the L_IDLE state, writes to this field shall have no effect.
27	0h RW	**Aggressive Slumber Partial (ASP):** When set to 1, and the ALPE bit is set, the HBA shall aggressively enter the Slumber state when it clears a bit in the PxCI or PxSACT register and the register values are then PxCI = 0h and PxSACT = 0h. When cleared, and the ALPE bit is set, the HBA will aggressively enter the Partial state when it clears a bit in the PxCI or PxSACT register and the register values are then PxCI = 0h and PxSACT = 0h. If CAP.SALP is cleard to 0., software shall treat this bit as reserved.
26	0h RW	**Aggressive Link Power Management Enable (ALPE):** When set, the HBA will aggressively enter a lower link power state (Partial or Slumber) based upon the setting of the ASP bit. Software shall only set this bit to 1 if CAP.SALP is set to 1. If CAP.SALP is cleared to 0, software shall treat this bit as reserved. BIOS is recommeded to program this field to 1.
25	0h RW	**Drive LED on ATAPI Enable (DLAE):** When set, the HBA will drive the LED pin active for commands regardless of the state of PxCMD.ATAPI. When cleared, the HBA will only drive the LED pin active for commands if PxCMD.ATAPI is set to 0.. This bit is set by software
24	0h RW	**Device is ATAPI (ATAPI):** When set, the connected device is an ATAPI device. This bit is used by the HBA to control whether or not to generate the desktop LED when commands are active.
23	0h RW	**Automatic Partial to Slumber Transitions Enabled (APSTE):** When set to 1., the HBA may perform Automatic Partial to Slumber Transitions. When cleared to 0. the port shall not perform Automatic Partial to Slumber Transitions. Software shall only set this bit to 1. if CAP2.APST is set to 1.; if CAP2.APST is cleared to 0. software shall treat this bit as reserved.
22	0h RO	**FIS-based Switching Capable Port (FBSCP):** The SATA controller does not support FIS-Based Switching.
21	0h RW/O	**External SATA Port (ESP):** When set to 1, indicates that this port is routed externally and will be used with an external SATA device. When set to 1, CAP.SXS must also be set to 1. When cleared (0), indicates that this port is not routed externally and supports internal SATA devices only. If ESP is set to '1', then the port may experience hot plug events.
20	0h RO	**Cold Presence Detection (CPD):** The SATA controller does not support cold presence detect.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW/O	**Mechanical Presence Switch Attached to Port (MPSP):** If set to 1, the platform supports a mechanical presence switch attached to this port. If cleared to 0, the platform does not support a mechanical presence switch attached to this port.
18	0h RW/O	**Hot Plug Capable Port (HPCP):** This indicates whether the this port is connected to a device which can be hot plugged. SATA by definition is hot-pluggable, but not all platforms are constructed to allow the device to be removed (it may be screwed into the chassis, for example). This bit can be used by system software to indicate a feature such as eject device to the end-user.
17	0h RW/V	**Port Multiplier Attached (PMA):** When set, a Port Multiplier is attached to the HBA for this port. When cleared, a Port Multiplier is not attached to the HBA for this port. This bit is a read only 0. when CAP.PMS = 0., and read/write when CAP.PMS = 1.. Note that this bit is set by software; hardware does not auto-detect that a Port Multiplier is attached.
16	0h RO	Reserved.
15	0h RO/V	**Command List Running (CR):** When this bit is set it indicates that the command list DMA engine for the port is running.
14	0h RO/V	**FIS Receive Running (FR):** When this bit is set it indicates that the FIS Receive DMA engine for the port is running. Note to software: When FR bit stays set, please read the PxIS.PCS and PxSERR.DET.X to service the PCS interrupt accordingly if any
13	0h RO/V	**Mechanical Presence Switch State (MPSS):** The MPSS bit reports the state of a mechanical presence switch attached to this port. If CAP.SMPS is set to 1 and the mechanical presence switch is closed then this bit is cleared to 0. If CAP.SMPS is set to 1 and the mechanical presence switch is open then this bit is set to 1. If CAP.SMPS is set to 0 then this bit is cleared to 0. Software should only use this bit if both CAP.SMPS and PxCMD.MPSP are set to 1.
12:8	0h RO/V	**Current Command Slot (CCS):** Indicates the current command slot the HBA is processing. This field is valid when the PxCMD.ST bit is set, and is constantly updated by the HBA. This field can be updated as soon as the HBA recognizes an active command slot, or at some point soon after when it begins processing the command. When PxCMD.ST transitions from a 1 to a 0, the HBA will reset this field to 0. After PxCMD.ST transitions from 0 to 1, the highest priority slot to issue from next is command slot 0. After the first command has been issued, the highest priority slot to issue from next is PxCMD.CCS + 1. For example, after the HBA has issued its first command, if PxCMD.CCS = 0h and PxCI is set to 3h, the next command that will be issued is from command slot 1.
7:5	0h RO	Reserved.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW	**FIS Receive Enable (FRE):** When set, the HBA may post received FISes into the FIS receive area pointed to by PxFB and PxFBU. When cleared, received FISes are not accepted by the HBA, except for the first D2H register FIS after the initialization sequence.
3	0h RW/1S	**Command List Override (CLO):** Setting this bit to 1 causes PxTFD.STS.BSY and PxTFD.STS.DRQ to be cleared to 0. This allows a software reset to be transmitted to the device regardless of whether the BSY and DRQ bits are still set in the PxTFD.STS register. The HBA sets this bit to 0 when PxTFD.STS.BSY and PxTFD.STS.DRQ have been cleared to 0. A write to this register with a value of 0 shall have no effect.
2	1h RO	**Power On Device (POD):** The SATA controller does not support cold presence detect.
1	0h RW/V	**Spin-Up Device (SUD):** This bit is read/write for HBAs that support staggered spin-up via CAP.SSS. This bit is read only 1. for HBAs that do not support staggered spin-up. On an edge detect from 0. to 1, the HBA shall start a COMRESET initializatoin sequence to the device. Clearing this bit causes no action on the interface. Clearing this bit to0 does not cause any OOB signal to be sent on the interface. When this bit is cleared to 0. and PxSCTL.DET = 0h, the HBA will enter listen mode.
0	0h RW	**Start (ST):** When set, the HBA may process the command list. When cleared, the HBA may not process the command list. Whenever this bit is changed from a 0 to a 1, the HBA starts processing the command list at entry 0. Whenever this bit is changed from a 1 ot a 0, the PxCI and PxSACT register is cleared by the HBA upon the HBA putting the controller into an idle state. PxTFD shall be updated also.

19.84 Port [0-7] Task File Data (PxTFD1)—Offset 1A0h

Port [0-7] Task File Data

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 7Fh

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	Reserved.
15:8	0h RO/V	**Error (ERR):** Contains the latest copy of the task file error register.
7	0h RO/V	**Status Busy (STS_BSY):** Status - Indicates the interface is busy.

Bit Range	Default & Access	Field Name (ID): Description
6:4	7h RO/V	**Status Rsvd0 (STS_RSVD0):** Status - Not Applicable.
3	1h RO/V	**Status Drq (STS_DRQ):** Status - Indicates a data transfer is requested.
2:1	3h RO/V	**Status Rsvd1 (STS_RSVD1):** Status - Not Applicable.
0	1h RO/V	**Status Err (STS_ERR):** Status - Indicates an error during the transfer.

19.85 Port [0-7] Signature (PxSIG1)—Offset 1A4h

Port [0-7] Signature

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: FFFFFFFFh

Bit Range	Default & Access	Field Name (ID): Description
31:0	FFFFFFFFh RO/V	**Signature (SIG):** Contains the signature received from a device on the first D2H Register FIS.

19.86 Port [0-7] Serial ATA Status (PxSSTS1)—Offset 1A8h

Port [0-7] Serial ATA Status

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	Reserved.
11:8	0h RO/V	**Interface Power Management (IPM):** Indicates the current interface state
7:4	0h RO/V	**Current Interface Speed (SPD):** Indicates the negotiated interface communication speed.
3:0	0h RO/V	**Device Detection (DET):** Indicates the interface device detection and Phy state.

19.87 Port [0-7] Serial ATA Control (PxSCTL1)—Offset 1ACh

Port [0-7] Serial ATA Control

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RO	Reserved.
19:16	0h RW	**Port Multiplier Port (PMP):** This field is not used by AHCI.
15:12	0h RW	**Select Power Management (SPM):** This field is not used by AHCI.
11:8	0h RW	**Interface Power Management Transitions Allowed (IPM):** Indicates which power states the HBA is allowed to transition to. If an interface power management state is not allowed via this register field, the HBA will not initiate that state and the HBA will PMNAKP any request from the device to enter that state.
7:4	0h RW	**Speed Allowed (SPD):** Indicates the highest allowable speed of the interface.
3:0	0h RW	**Device Detection Initialization (DET):** Controls the HBA.s device detection and interface initialization.

19.88 Port [0-7] Serial ATA Error (PxSERR1)—Offset 1B0h

Port [0-7] Serial ATA Error

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/1C/V	**Diagnostics (DIAG):** Contains diagnostic error information for use by diagnostic software in validating correct operation or isolating failure modes.
15:0	0h RW/1C/V	**Error (ERR):** The ERR field contains error information for use by host software in determining the appropriate response to the error condition. If one or more of bits 11:8 of this register are set, the controller will stop the current transfer.

19.89 Port [0-7] Serial ATA Active (PxSACT1)—Offset 1B4h

Port [0-7] Serial ATA Active

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/1S/V	**Device Status (DS):** System software sets this bit for native command queuing commands prior to setting the PxCI.CI bit in the same command slot entry. This field is cleared via the Set Device Bits FIS.

19.90 Port [0-7] Commands Issued (PxCI1)—Offset 1B8h

Port [0-7] Commands Issued

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/1S/V	**Commands Issued (CI):** This field is set by software to indicate to the HBA that a command has been built in system memory for a command slot and may be sent to the device. When the HBA receives a FIS which clears the BSY, ERR, and DRQ bits for the command, it clears the corresponding bit in this register for that command slot. Bits in this field shall only be set to 1 by software when PxCMD.ST is set to 1.

19.91 Port [0-7] SNotification (PxSNTF1)—Offset 1BCh

Port [0-7] SNotification

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	Reserved.
15:0	0h RW/1C/V	**PM Notify (PMN):** This field indicates whether a particular device with the corresponding PM Port number issued a Set Device Bits FIS to the host with the Notification bit set. PM Port 0h sets bit 0. PM Port Fh sets bit 15. Individual bits are cleared by software writing 1's to the corresponding bit positions. Note that, while this field is reset to default on a HBA Reset, it is not reset by COMRESET or SRST

19.92 Port [0-7] Device Sleep (PxDEVSLP1)—Offset 1C4h

Port [0-7] Device Sleep

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 1E022852h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	Reserved.
28:25	Fh RW/V	**DITO Multiplier (DM):** 0's based value that specifies the DITO multiplier that the HBA applies to the specified DITO value, effectively extending the range of DITO from 1ms to 16368ms. A value of 0h indicates a multiplier of 1. A maximum multiplier of 16 may be applied. The HBA computes the total idle timeout as a product of DM and DITO (i.e. DITO actual = DITO * DM).
24:15	4h RW/V	**DEVSLP Idle Timeout (DITO):** This field specifies the amount of the time (in approximate 1ms granularity) that the HBA shall wait before driving the DEVSLP signal.Hardware reloads its port specific DEVSLP timer with this value each time the port transitions out of DEVSLP state. For example: from DEVSLP to active or PxDEVSLP.ADSE transitions from 0 to a 1. If CAP2.SDS or CAP2.SADM or PxDEVSLP.DSP is cleared to 0 then these bits are read-only 0h and software shall treat these bits as reserved. Software shall only set this value when PxCMD.ST is cleared to 0 and PxDEVSLP.ADSE is cleared to 0.

Bit Range	Default & Access	Field Name (ID): Description
14:10	Ah RW/V	**DEVSLP Minimum Assertion Time (MDAT):** This field specifies the minimum amount of time (in 1ms granularity) that the HBA must assert the DEVSLP signal before it may be de-asserted. The nominal value is 10ms and the minimum is 1ms depending on device identification information. If CAP2.SDS is cleared to 0 or PxDEVSLP.DSP is cleared to 0 then these bits are read-only 0h and software shall treat these bits as reserved. Software shall only set this value when PxCMD.ST is cleared to 0, PxDEVSLP.ADSE is cleared to 0, and prior to setting PxCMD.ICC to 8h.
9:2	14h RW/V	**DEVSLP Exit Timeout (DETO):** This field specifies the maximum duration (in approximate 1ms granularity) from DEVSLP de-assertion until the device is ready to accept OOB. The nominal value is 20ms while the max value is 255ms depending on device identification information. If CAP2.SDS is cleared to 0 or PxDEVSLP.DSP is cleared to 0 then these bits are read-only 0h and software shall treat these bits as reserved. Software shall only set this value when PxCMD.ST is cleared to 0, PxDEVSLP.ADSE is cleared to 0, and prior to setting PxCMD.ICC to 8h.
1	1h RW/O	**DEVSLP Present (DSP):** If set to '1', the platform supports DEVSLP on this port. If cleared to '0', the platform does not support DEVSLP on this port. This bit may only be set to '1' if CAP2.SDS is set to '1'.DSP is mutually exclusive with the PxCMD.HPCP bit and PxCMD.ESP bit. Note that these bits are not reset on a HBA reset.
0	0h RW/V	**Aggressive DEVSLP Enable (ADSE):** This bit is read/write for HBAs that support aggressive DEVSLP management (CAP2. SADM = 1). When this bit is set to 1, the HBA shall assert the DEVSLP signal after the port has been idle (PxCI = 0h and PxSACT = 0h) for the amount of time specified by the PxDEVSLP.DITO register and the interface is in Slumber (PxSSTS.IPM = 6h). When this bit is cleared to 0, the HBA does not enter DEVSLP unless software directed via PxCMD.ICC. This bit shall only be set to 1 if PxDEVSLP.DSP is set to 1. If this bit is set to 1 and software clears the bit to 0, then the HBA shall de-assert the DEVSLP signal if asserted. Note that these bits are not reset on a HBA reset. BIOS is recommended to program this field to 1 if the platform support the DEVSLP feature. If CAP2.SDS is cleared to 0 or CAP2.SADM is cleared to 0, or if PxDEVSLP.DSP is cleared to 0 then these bits are read-only 0h and software shall treat these bits as reserved.

19.93 Enclosure Management Message Format (EM_MF)—Offset 580h

Enclosure Management Message Format

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	Reserved.
27:24	0h RW	**Message Type (MTYPE):** Specifies the type of the message. The message types are: • 0h: LED. • 1h: SAF-TE. • 2h: SES-2. • 3h: SGPIO (register based interface). • All other values reserved.
23:16	0h RW	**Data Size (DSIZE):** Specifies the data size in bytes. If the message (enclosure services command) has a data buffer that is associated with it that is transferred, the size of that data buffer is specified in this field. If there is no separate data buffer, this field shall have a value of '0'. The data directly follows the message in the message buffer. This value should always be '0'.
15:8	0h RW	**Message Size (MSIZE):** Specifies the size of the message in bytes. The message size does not include the one Dword header. A value of '0'. is invalid. The message size is always 4 bytes.
7:0	0h RO	Reserved.

19.94 Enclosure Management LED (EM_LED)—Offset 584h

Enclosure Management LED

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Value (VAL):** This field describes the state of each LED for a particular location. There are three LEDs that may be supported by the HBA. Each LED has 3 bits of control. LED values are: 000b - LED shall be off; 001b - LED shall be solid on as perceived by human eye; All other values reserved. The LED bit locations are: Bits 2:0 - Activity LED (may be driven by hardware); Bits 5:3 - Vendor Specific LED (e.g. locate); Bits 8:6 - Vendor Specific LED (e.g. fault); Bits 15:9 - Reserved. Vendor specific message is: Bit 3:0 - Vendor Specific Pattern; Bit 15:4 - Reserved. Note: If Activity LED Hardware Driven (ATTR.ALHD) bit is set, host will output the hardware LED value sampled internally and will ignore software written activity value on bit [2:0]. Since Enclosure Management does not support port multiplier based LED message, the LED message will be generated independently based on respective port's operation activity. Vendor specific LED values Locate (Bits 5:3) and Fault (Bits 8:6) always are driven by software..
15:8	0h RW	**Port Multiplier Information (PM):** Specifies slot specific information related to Port Multiplier. Bits 3:0 specify the Port Multiplier port number for the slot that requires the status update. If a Port Multiplier is not attached to the device in the affected slot, the Port Multiplier port number shall be '0'. Bits 7:4 are reserved. SATA does not support LED messages for devices behind a Port MUltiplier. This byte should be 0.
7:0	0h RW	**HBA Information (HBA):** Specifies slot specific information related to the HBA. Bits 4:0 - HBA port number for the slot that requires the status update. Bit 5 - If set to '1', Value is a vendor specific message that applies to the entire enclosure. If cleared to '0', Value applies to the port specified in bits 4:0. Bits 7:6 - Reserved.

19.95 MSI-X Pending Bit Array QW 0 (MXPQW0_DW0)—Offset 0h

MSI-X Pending Bit Array QW 0

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	Reserved.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO/V	**MSI-X Vector Pending (MXVP):** For each Pending Bit that is set, the function has a pending message for the associated MSI-X Table entry. Pending bits that have no associated MSI-X Table entry are reserved. After reset, the state of reserved Pending bits must be 0. Software should never write, and should only read Pending Bits. If software writes to Pending Bits, the result is undefined. Each Pending Bit's state after reset is 0 (no message pending).

19.96 MSI-X Pending Bit Array QW 1 (MXPQW0_DW1)—Offset 4h

MSI-X Pending Bit Array QW 1

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**MSI-X Vector Pending QW1 (Rsvd):** For each Pending Bit that is set, the function has a pending message for the associated MSI-X Table entry. Pending bits that have no associated MSI-X Table entry are reserved. After reset, the state of reserved Pending bits must be 0. Software should never write, and should only read Pending Bits. If software writes to Pending Bits, the result is undefined. Each Pending Bit's state after reset is 0 (no message pending).

19.97 MSI-X Table Entries 0 Message Lower Address (MXTE0MLA)—Offset 0h

MSI-X Table Entries 0 Message Lower Address

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**MSI-X Message Lower Address (MXMLA):** Specifies the lower 32-bit of the DWORD-aligned address (Addr[31:02]) of the MSI-X Message.
1:0	0h RO	Reserved.

19.98 MSI-X Table Entries 0 Message Upper Address (MXTE0MUA)—Offset 4h

MSI-X Table Entries 0 Message Upper Address

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**MSI-X Message Upper 32-Bit Address (MXMUA):** Specifies the lower 32-bit of the DWORD-aligned address (Addr[31:02]) of the MSI-X Message.

19.99 MSI-X Table Entries 0 Message Data (MXTE0MD)—Offset 8h

MSI-X Table Entries 0 Message Data

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**MSI-X Message Data (MXMD):** Specifies the 32-bit Data of the MSI-X Message.

19.100 MSI-X Table Entries 0 Vector Control (MXTE0VC)—Offset Ch

MSI-X Table Entries 0 Vector Control

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	Reserved.

Bit Range	Default & Access	Field Name (ID): Description
0	1h RW	**MSI-X Vector Mask (MXVM):** When this bit is set, the function is prohibited from sending a message using this MSI-X Table entry. However, any other MSI-X Table entries programmed with the same vector will still be capable of sending an equivalent message unless they are also masked. This bit's state after reset is 1 (entry is masked).

§ §

20 I/O Sensor Hub (ISH) Registers

This chapter documents the registers in Bus: 0, Device 17, Function 0.

20.1 DEVICEVENDORID - Device ID and Vendor ID Register (DEVVENDID) — Offset 0h

Device ID and Vendor ID provided by this register uniquely identifies the particular Device

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + 0h	22D88086 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	22D8h RO	**DEVICEID:** Device ID identifies the particular PCI device
15:0	8086h RO	**VENDORID:** Vendor ID is a unique ID provided by the PCI SIG, which identifies the manufacturer of the device

20.2 STATUSCOMMAND — Offset 4h

Command register to programme interrupt disable , bus master enable, and Memory space enable. Status register to read the errors and aborts

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + 4h	100000 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**RESERVED0:** Reserved
29	0h RW/1C	**RMA:** Received Master Abort
28	0h RW/1C	**RTA:** Received Target Abort
27:21	0h RO	**RESERVED1:** Reserved
20	1h RO	**CAPLIST:** Capabilities List: Indicates that the controller contains a capabilities pointer list
19	0h RO	**INTR_STATUS:** Interrupt Status: This bit reflects state of interrupt in the device

Bit Range	Default & Access	Field Name (ID): Description
18:16	0h RO	**RESERVED2:** Reserved
15:11	0h RO	**RESERVED3:** Reserved
10	0h RW	**INTR_DISABLE:** Interrupt Disable
9	0h RO	**RESERVED4:** Reserved
8	0h RW	**SERR_ENABLE:** SERR Enable , Not implemented
7:3	0h RO	**RESERVED5:** Reserved
2	0h RW	**BME:** Bus Master Enable
1	0h RW	**MSE:** Memory Space Enable
0	0h RO	**RESERVED6:** Reserved

20.3 REVCLASSCODE — Offset 8h

Revision ID register identifies revision of particular device and Class Code register is used to identify generic function of the device

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + 8h	6 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**CLASS_CODES:** Class Code register is read-only and is used to identify the generic function of the device, and in some cases, a specific register-level programming interface
7:0	6h RO	**RID:** Revision ID identifies the revision of particular PCI device.

20.4 CLLATHEADERBIST — Offset Ch

Cache Line size as RW with def 0, Latency timer RW with def 0, Header type with Type 0 configuration header and Reserved BIST register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**RESERVED0:** Reserved
23	0h RO	**MULFNDEV:** Multi-Function Device
22:16	0h RO	**HEADERTYPE:** Header Type: Implements Type 0 Configuration header
15:8	0h RO	**LATTIMER:** Latency Timer:. This register is implemented as R/W with default as 0
7:0	0h RW	**CACHELINE_SIZE:** Cacheline Size

20.5 BAR — Offset 10h

Base Address Register low [31:2] , type[2:1] in 32bit or 64bit addr range and memory space indicator [0]

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + 10h	4 h

Bit Range	Default & Access	Field Name (ID): Description
31:13	0h RW	**BASEADDR:** Base Address Register Low, Base address of the OCP fabric memory space. Taken from Strap values as ones
12:4	0h RO	**SIZEINDICATOR:** Size Indicator RO Always returns 0 The size of this register depends on the size of the memory space
3	0h RO	**PREFETCHABLE:** Prefetchable: Indicates that this BAR is not prefetchable
2:1	2h RO	**TYPE:** If BAR_64b_EN is 0 then 00 indicates BAR lies in 32bit address range, If BAR_64b_EN is 1 then 10 Indicates BAR lies in 64 bit address range
0	0h RO	**MESSAGE_SPACE:** Memory Space Indicator: 0 indicates this BAR is present in the memory space.

20.6 BAR -Base Address Register High (BAR_HIGH) — Offset 14h

Base Address Register High enabled if [2:1] of BAR_type_LOW is 10

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + 14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**BASEADDR_HIGH:** Base Address high - MSB

20.7 BAR1 — Offset 18h

Base Address Register1 accesses to PCI configuration space and is always 4K, type in [2:1] and memory space indicator in [0]

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + 18h	4 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**BASEADDR1:** Base Address1 This field is present if BAR1 is enabled through private configuration space.
11:4	0h RO	**SIZEINDICATOR1:** Always is 0 as minimum size is 4K
3	0h RO	**PREFETCHABLE1:** Prefetchable: Indicates that this BAR is not prefetchable.
2:1	2h RO	**TYPE1:** If BAR_64b_EN is 0 then 00 indicates BAR lies in 32bit address range If BAR_64b_EN is 1 then 10 Indicates BAR lies in 64 bit address range
0	0h RO	**MESSAGE_SPACE1:** Memory Space Indicator: 0 Indicates this BAR is present in the memory space

20.8 BAR1 -Base Address Register1 High (BAR1_HIGH) — Offset 1Ch

Base Address Register1 High enabled only if [2:1] of BAR1 register is 10

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + 1Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**BASEADDR1_HIGH:** Base Address: Base address of the OCP fabric memory space. Taken from Strap values as ones

20.9 SUBSYSTEMID — Offset 2Ch

SVID register along with SID register is to distinguish subsystem from another

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**SUBSYSTEMID:** Subsystem ID: This register is implemented for any function that can be instantiated more than once in a given system.
15:0	0h RW/O	**SUBSYSTEMVENDORID:** Subsystem Vendor ID: This register must be implemented for any function that can be instantiated more than once in a given system

20.10 EXPANSION ROM base address (EXPANSION_ROM_BASEADDR) — Offset 30h

EXPANSION ROM base address register is a RO indicates support for expansion ROMs

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + 30h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**EXPANSION_ROM_BASE:** Expansion ROM Base Address: Value of all zeros indicates no support for Expansion ROM

20.11 CAPABILITYPTR — Offset 34h

Capabilities Pointer register indicates what the next capability is

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + 34h	80 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**RESERVED0:** Reserved
7:0	80h RO	**CAPPTR_POWER:** Capabilities Pointer: Indicates what the next capability is.

20.12 INTERRUPTREG — Offset 3Ch

Interrupt line Register isn't used in Bridge directly, Interrupt Pin register reflects the IPIN value in private config space. Min_gnt register indicating the req of latency timers and max_lat register max latency

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + 3Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**MAX_LAT:** Value of 0 indicates device has no major requirements for the settings of latency timers
23:16	0h RO	**MIN_GNT:** Value of 0 indicates device has no major requirements for the settings of latency timers
15:12	0h RO	**RESERVED0:** Reserved
11:8	0h RO	**INTPIN:** Interrupt Pin: Value in this register is reflected from the IPIN value in the private configuration space.
7:0	0h RW	**INTLINE:** Interrupt Line: It is used to communicate to software, the interrupt line to which the interrupt pin is connected

20.13 POWERCAPID — Offset 80h

PowerManagement Capability ID register points to next capability structure and power mgmnt capability , with Power management capabilities register for PME support and version

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + 80h	39001 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	0h RO	**PMESUPPORT:** This 5-bit field indicates the power states in which the function can assert the PME#
26:19	0h RO	**RESERVED0:** Reserved
18:16	3h RO	**VERSION:** Indicates support for Revision 1.2 of the PCI Power Management Specification
15:8	90h RO	**NXTCAP:** Next Capability: Points to the next capability structure.
7:0	1h RO	**POWER_CAP:** Power Management Capability: Indicates this is power management capability

20.14 PMECTRLSTATUS — Offset 84h

power management control and status register to set and read PME status, PME enable, No Soft reset and power state

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + 84h	8 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**RESERVED0:** Reserved
15	0h RW/1C	**PMESTATUS:** PME Status
14:9	0h RO	**RESERVED1:** Reserved
8	0h RW	**PMEENABLE:** PME Enable
7:4	0h RO	**RESERVED2:** Reserved
3	1h RO	**NO_SOFT_RESET:** This bit indicates that devices transitioning from D3hot to D0 because of Powerstate commands do not perform an internal reset
2	0h RO	**RESERVED3:** Reserved
1:0	0h RW	**POWERSTATE:** Power State: This field is used both to determine the current power state and to set a new power state

20.15 PCI DEVICE IDLE CAPABILITY RECORD (PCIDEVIDLE_CAP_RECORD) — Offset 90h

PCI Device Vendor Specific Capability register defines Vendor specific Capability ID, revision , length , next capability and CAPID

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + 90h	F0140009 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RO	**VEND_CAP:** Vendor Specific Capability ID
27:24	0h RO	**REVID:** Revision ID of capability structure
23:16	14h RO	**CAP_LENGTH:** Vendor Specific Capability Length
15:8	0h RO	**NEXT_CAP:** Next Capability
7:0	9h RO	**CAPID:** Capability ID

20.16 DEVID VENDOR SPECIFIC REG (DEVID_VEND_SPECIFIC_REG) — Offset 94h

Extended Vendor capability register for VSEC Length, revision and ID

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + 94h	1400010 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	14h RO	**VSEC_LENGTH:** Vendor Specific Extended Capability Length
19:16	0h RO	**VSEC_REV:** Vendor specific Extended Capability revision
15:0	10h RO	**VSECID:** Vendor Specific Extended Capability ID

20.17 D0I3_CONTROL_SW_LTR_MMIO_REG — Offset 98h

Software location pointer in MMIO space as an offset specified by BAR

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + 98h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**SW_LAT_DWORD_OFFSET:** SW LTR Update MMIO Offset Location (SWLTRLOC)
3:1	0h RO	**SW_LAT_BAR_NUM:** Indicates that the SW LTR update MMIO location is always at BAR0
0	0h RO	**SW_LAT_VALID:** This value is reflected from the SW LTR valid strap at the top level

20.18 DEVICE_IDLE_POINTER_REG — Offset 9Ch

Device IDLE pointer register giving details on Device MMIO offset location , BAR NUM and D0i3 Valid Strap

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + 9Ch	6D01 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	6D0h RO	**DWORD_OFFSET:** contains the location pointer to the SW LTR register in MMIO space, as an offset from the specified BAR
3:1	0h RO	**BAR_NUM:** Bar num: Indicates that the D0i3 MMIO location is always at BAR0
0	1h RO	**VALID:** Valid: This value is reflected from the D0i3 valid strap at the top level.

20.19 D0I3_MAX_POW_LAT_PG_CONFIG — Offset A0h

D0idle_Max_Power_On_Latency register set at boot and Power control enable register to enable communication with the PGCB block below the Bridge

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + A0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**RESERVED0:** Reserved
21	0h RO	**HAE:** Hardware Autonomous Enable
20	0h RO	**RESERVED1:** Reserved
19	0h RO	**SLEEP_EN:** Sleep Enable
18	0h RO	**PGE:** DEVIDLE Enable (DEVIDLEN): If ?1?, then the function will power gate when idle and the DevIdle register (DevIdleC[2] = ?1?) is set.
17	0h RO	**I3_ENABLE:** D3-Hot Enable (D3HEN): If ?1?, then function will power gate when idle and the PMCSR[1:0] register in the function =?11? (D3).
16	0h RO	**PMCRE:** PMCRE: PMC Request Enable
15:13	0h RO	**RESERVED2:** Reserved
12:10	0h RW/O	**POW_LAT_SCALE:** Power On Latency Scale
9:0	0h RW/O	**POW_LAT_VALUE:** Power On Latency value

20.20 GEN_REGRW1 - General Purpose Read Write Register1 (GEN_REGRW1) — Offset B0h

General Purpose PCI Read Write Register1

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + B0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW1:** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg1 Out of Band signal

20.21 GEN_REGRW2 — Offset B4h

General Purpose PCI Read Write Register2

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + B4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW2:** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg2 Out of Band signal

20.22 GEN_REGRW3 — Offset B8h

General Purpose PCI Read Write Register3

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + B8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW3:** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg3 Out of Band signal

20.23 GEN_REGRW4 — Offset BCh

General Purpose PCI Read Write Register4

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + BCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW4:** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg4 Out of Band signal

20.24 GEN_INPUT_REG — Offset C0h

General Purpose Input Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + C0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**GEN_REG_INPUT_RW:** General Purpose Input Register: This register value reflects the value of oob_gen_input_pci Out of Band signal

20.25 MANID — Offset F8h

Manufacturers ID register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:17, F:0] + F8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**MANID:** Manufacturer ID: Default value comes from straps.

20.26 Peripheral Interrupt Status (PISR)—Offset 1000h

This register contains all the inbound and outbound interrupt status bits of the IPC registers. Interrupts ISH2HOST_Intr, ISH2SEC_Intr, ISH2PMC_Intr, HOST2ISH_intr, SEC2ISH_Intr and PMC2ISH_Intr are generated when PISR[x] & PIMR[x].

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:27	0h RO	**Reserved (RESERVED1):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
26	0h RO	**Interrupt Status Bit For Audio CSR Register (PISR_AUDIO_CSR):** Audio to ISH CSR Ored interrupt: • 1: Interrupt is active. • 0: Interrupt is inactive.
25	0h RO	**Interrupt Status Bit for ISP CSR Register (PISR_ISP_CSR):** ISP to ISH CSR Ored interrupt: • 1: Interrupt is active. • 0: Interrupt is inactive.
24	0h RO	**Interrupt Status Bit For PMC CSR Register (PISR_PMC_CSR):** PMC to ISH CSR Ored interrupt: • : Interrupt is active. • 0: Interrupt is inactive.
23	0h RO	**Interrupt Status Bit For CSME CSR Register (PISR_CSME_CSR):** CSME to ISH CSR Ored interrupt: • 1: Interrupt is active. • 0: Interrupt is inactive.
22:12	0h RO	**Reserved (RESERVED0):** Reserved.
11	0h RO	**ISH to Audio Status (PISR_ISH2AUDIO):** Outbound IPC request from ISH to audio status: • 1: Interrupt is active. • 0: Interrupt is inactive.
10	0h RO	**ISH to ISP Status (PISR_ISH2ISP):** Outbound IPC request from ISH to ISP status: • 1: Interrupt is active. • 0: Interrupt is inactive.
9	0h RO	**ISH to GFX Status (PISR_ISH2GFX):** Outbound IPC request from ISH to GFX status: • 1: Interrupt is active. • 0: Interrupt is inactive.
8	0h RO	**Audio to ISH Status (PISR_AUDIO2ISH):** Inbound IPC request from audio to ISH status: • 1: Interrupt is active. • 0: Interrupt is inactive.
7	0h RO	**ISP to ISH Status (PISR_ISP2ISH):** Inbound IPC request from ISP to ISH status: • 1: Interrupt is active. • 0: Interrupt is inactive.
6	0h RO	**GFX to ISH Status (PISR_GFX2ISH):** Inbound IPC request from GFX to ISH status: • 1: Interrupt is active. • 0: Interrupt is inactive.
5	0h RO	**ISH to SEC Status (PISR_ISH2SEC):** Outbound IPC request from ISH to SEC status: • 1: Interrupt is active. • 0: Interrupt is inactive.
4	0h RO	**ISH to PMC Status (PISR_ISH2PMC):** Outbound IPC request from ISH to PMC status: • 1: Interrupt is active. • 0: Interrupt is inactive.
3	0h RO	**Reserved (RESERVED):** Reserved.
2	0h RO	**SEC to ISH Status (PISR_SEC2ISH):** Inbound IPC request from SEC to ISH status: • 1: Interrupt is active. • 0: Interrupt is inactive.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RO	**PMC to ISH Status (PISR_PMC2ISH):** Inbound IPC request from PMC to ISH status: • 1: Interrupt is active. • 0: Interrupt is inactive.
0	0h RO	**Host to ISH Status (PISR_HOST2ISH):** Inbound IPC request from host to ISH status: • 1: Interrupt is active. • 0: Interrupt is inactive.

20.27 Peripheral Interrupt Mask (PIMR)—Offset 1004h

This register enables or disables Inbound and outbound interrupts between ISH&HOST, ISH&PMC and ISH&PMC. The outbound interrupts cannot be masked by HOST, PMC & SEC since this register is not accessible by external agents like HOST, PMC & SEC.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RESERVED1):** Reserved.
27	0h RW	**H2IBCISC Interrupt Mask (H2IBCISC_IE):** Mask bit for H2IBCISC interrupt mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
26	0h RW	**Audio CSR Interrupt Mask (PIMR_AUDIO_CSR_MASK):** Mask bit for audio CSR interrupt mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
25	0h RW	**ISP CSR Interrupt Mask (PIMR_ISP_CSR_MASK):** Mask bit for ISP CSR interrupt mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
24	0h RW	**PMC CSR Interrupt Mask (PIMR_PMC_CSR_MASK):** Mask bit for PMC CSR interrupt mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
23	0h RW	**CSME CSR Interrupt Mask (PIMR_CSME_CSR_MASK):** Mask bit for CSME CSR interrupt mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
22	0h RW	**ISH to Audio Busy Clear Mask (PIMR_ISH2AUDIO_BUSY_CLEAR):** Mask bit for ISH to audio busy clear interrupt: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
21	0h RW	**ISH to ISP Busy Clear Mask (PIMR_ISH2ISP_BUSY_CLEAR):** Mask bit for ISH to ISP busy clear interrupt: • 1: Interrupt is unmasked. • 0: Interrupt is masked.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**ISH2 to GFX Busy Clear Mask (PIMR_ISH2GFX_BUSY_CLEAR):** Mask bit for ISH to GFX busy clear interrupt: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
19	0h RW	**ISH to Audio Mask (PIMR_ISH2AUDIO):** Outbound IPC request from ISH to audio mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
18	0h RW	**ISH to ISP Mask (PIMR_ISH2ISP):** Outbound IPC request from ISH to ISP mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
17	0h RW	**ISH to GFX Mask (PIMR_ISH2GFX):** Outbound IPC request from ISH to GFX mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
16	0h RW	**Audio to ISH Mask (PIMR_AUDIO2ISH):** Inbound IPC request from audio to ISH mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
15	0h RW	**ISP to ISH Mask (PIMR_ISP2ISH):** Inbound IPC request from ISP to ISH mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
14	0h RW	**GFX to ISH Mask (PIMR_GFX2ISH):** Inbound IPC request from GFX to ISH mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
13	0h RW	**ISH to SEC Busy Clear Mask (PIMR_ISH2SEC_BUSY_CLEAR):** Mask bit for ISH to SEC busy clear interrupt: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
12	0h RW	**ISH to PMC Busy Clear Mask (PIMR_ISH2PMC_BUSY_CLEAR):** Mask bit for ISH to PMC busy clear interrupt: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
11	0h RW	**ISH to Host Busy Clear Mask (PIMR_ISH2HOST_BUSY_CLEAR):** Mask bit for ISH to host busy clear interrupt: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
10:6	0h RO	**Reserved (RESERVED0):** Reserved.
5	0h RW	**ISH to SEC Mask (PIMR_ISH2SEC):** Outbound IPC request from ISH to SEC mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
4	0h RW	**ISH to PMC Mask (PIMR_ISH2PMC):** Outbound IPC request from ISH to PMC mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
3	0h RO	**Reserved (RESERVED):** Reserved.
2	0h RW	**SEC to ISH Mask (PIMR_SEC2ISH):** Inbound IPC request from SEC to ISH mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**PMC to ISH Mask (PIMR_PMC2ISH):** Inbound IPC request from PMC to ISH mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
0	0h RW	**Host to ISH Mask (PIMR_HOST2ISH):** Inbound IPC request from host to ISH mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.

20.28 Host Peripheral Interrupt Mask (HOST_PIMR)—Offset 1008h

This register is for masking the interrupt, caused by setting the DB bit, for IPC from mIA to host.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Reserved (RESERVED1):** Reserved.
8	0h RW	**Host to ISH Busy Clear Interrupt Mask Bit (PIMR_HOST2ISH_BUSY_CLEAR):** • 1: Interrupt is unmasked. • 0: Interrupt is masked.
7:1	0h RO	**Reserved (RESERVED0):** Reserved.
0	0h RW	**Mask Interrupt Caused By ISH to Host Doorbell (PIMR_ISH2HOST_IPC_REG):** Outbound IPC request from ISH to host mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.

20.29 HOST Peripheral Interrupt Status (HOST_PISR)—Offset 100Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Reserved (RESERVED1):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW/1C	**Host to ISH Busy Clear Interrupt Status Bit (PISR_HOST2ISH_BUSY_CLEAR):** 1: Interrupt is active.
7:1	0h RO	**Reserved (RESERVED0):** Reserved.
0	0h RO	**ISH to Host DB Interrupt Status Bit (PISR_ISH2HOST_IPC_REG)**

20.30 Reserved (RESERVED2)—Offset 1010h

Reserved.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**Reserved (RESERVED2):** Reserved.

20.31 Reserved (RESERVED3)—Offset 1014h

Reserved.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**Reserved (RESERVED3):** Reserved.

20.32 Reserved (RESERVED4)—Offset 1018h

Reserved.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**Reserved (RESERVED4):** Reserved.

20.33 Reserved (RESERVED5)—Offset 101Ch

Reserved.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**Reserved (RESERVED5):** Reserved.

20.34 Reserved (RESERVED6)—Offset 1020h

Reserved.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**Reserved (RESERVED6):** Reserved.

20.35 Reserved (RESERVED7)—Offset 1024h

Reserved.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**Reserved (RESERVED7):** Reserved.

20.36 Reserved (RESERVED8)—Offset 1028h

Reserved.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**Reserved (RESERVED8):** Reserved.

20.37 Reserved (RESERVED9)—Offset 102Ch

Reserved.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**Reserved (RESERVED9):** Reserved.

20.38 Reserved (RESERVED10)—Offset 1030h

Reserved.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**Reserved (RESERVED10):** Reserved.

20.39 ISH Host Firmware Status (ISH_HOST_FWSTS)—Offset 1034h

This is a 32 bit register which is set to 0x0000 on a system reset or a wakeup from D3Cold. As the firmware comes up after the reset or power cycle it sets bits of this register to 1`b1 to indicate its status.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**ISH Host Firmware Status (ISH_HOST_FWSTS)**

20.40 Host Communication (HOST_COMM)—Offset 1038h

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Host Communication Register (HOST_COMM)**

20.41 ISH SEC Firmware Status (ISH_SEC_FWSTS)—Offset 103Ch

This is a 32 bit register which is set to 0x0000 on a system reset or a wakeup from D3Cold. As the firmware comes up after the reset or power cycle it sets bits of this register to 1`b1 to indicate its status.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**ISH SEC Firmware Status (ISH_SEC_FWSTS)**

20.42 SEC Communication (SEC_COMM)—Offset 1040h

This is a 32 bit register which is set to 0x0000 on a system reset or a wakeup from D3Cold. The SEC sets bits of this register to 1`b1 to communicate with the ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SEC Communication (SEC_COMM):** This is a 32 bit register which is set to 0x0000 on a system reset or a wakeup from D3Cold. The SEC sets bits of this register to 1`b1 to communicate with the ISH.

20.43 ISH Reset (ISH_RST)—Offset 1044h

This register is reserved for SEC; SEC should not write to this register. The reset register is a 32 bit register that can be written by ISH Minute IA. The host cannot read or write this bit. Setting bit 1 will cause MinuteIA to reset.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	**Reserved (RESERVED):** Reserved.
0	0h RW	**Reset Bit (RESET_BIT):** ISH reset register: • 1: Reset active. • 0: Reset inactive.

20.44 Inbound Doorbell Host to ISH (HOST2ISH_DOORBELL)—Offset 1048h

Inbound doorbell register, host core to interrupt ISH. Data 30:0 is 31 bit message payload used for backward compatibility.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** When this bit is cleared, the ISH CPU is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** 31 bit message payload for backward compatibility.

20.45 Inbound Doorbell PMC To ISH (PMC2ISH_DOORBELL)—Offset 104Ch

Inbound doorbell register. PMC to interrupt ISH Data 30:0 is 31 bit message payload used for backward compatibility.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** When this bit is cleared, the ISH CPU is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** 31 bit message payload for backward compatibility.

20.46 Inbound Doorbell SEC to ISH (SEC2ISH_DOORBELL)—Offset 1050h

Inbound doorbell register, SEC to interrupt ISH Data 30:0 is 31 bit message payload used for backward compatibility.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** When this bit is cleared, the ISH CPU is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** 31 bit message payload for backward compatibility.

20.47 Outbound Doorbell ISH To Host (ISH2HOST_DOORBELL)—Offset 1054h

Outbound doorbell register for the ISH to interrupt the host. Setting bit 31 of this register causes the host to receive a IRQn interrupt. Data 30:0 is 31 bit message payload used for backward compatibility.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** When this bit is cleared, the host CPU is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** 31 bit message payload for backward compatibility.

20.48 Outbound Doorbell ISH to PMC (ISH2PMC_DOORBELL)—Offset 1058h

Outbound doorbell register for the ISH to interrupt the PMC. Setting bit 31 of this register causes the PMC to receive interrupt on OOB. Data 30:0 is 31 bit message payload used for backward compatibility.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** When this bit is cleared, the PMC CPU is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** 31 bit message payload for backward compatibility.

20.49 Outbound Doorbell ISH to SEC (ISH2SEC_DOORBELL)—Offset 105Ch

Outbound doorbell register for the ISH to interrupt the SEC. Setting bit 31 of this register causes the SEC to receive interrupt on OOB Data 30:0 is 31 bit message payload used for backward compatibility.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** When this bit is cleared, the SEC CPU is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** 31 bit message payload for backward compatibility.

20.50 Outbound Inter Processor Messages 1 ISH to Host (ISH2HOST_MSG1)—Offset 1060h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.51 Outbound Inter Processor Messages 2 ISH to Host (ISH2HOST_MSG2)—Offset 1064h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.52 Outbound Inter Processor Messages 3 ISH to Host (ISH2HOST_MSG3)—Offset 1068h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.53 Outbound Inter Processor Messages 4 ISH to Host (ISH2HOST_MSG4)—Offset 106Ch

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.54 Outbound Inter Processor Messages 5 ISH to Host (ISH2HOST_MSG5)—Offset 1070h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.55 Outbound Inter Processor Messages 6 ISH to Host (ISH2HOST_MSG6)—Offset 1074h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.56 Outbound Inter Processor Messages 7 ISH to Host (ISH2HOST_MSG7)—Offset 1078h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.57 Outbound Inter Processor Messages 8 ISH to Host (ISH2HOST_MSG8)—Offset 107Ch

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.58 Outbound Inter Processor Messages 9 ISH to Host (ISH2HOST_MSG9)—Offset 1080h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.59 Outbound Inter Processor Messages 10 ISH to Host (ISH2HOST_MSG10)—Offset 1084h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.60 Outbound Inter Processor Messages 11 ISH to Host (ISH2HOST_MSG11)—Offset 1088h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.61 Outbound Inter Processor Messages 12 ISH to Host (ISH2HOST_MSG12)—Offset 108Ch

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.62 Outbound Inter Processor Messages 13 ISH to Host (ISH2HOST_MSG13)—Offset 1090h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.63 Outbound Inter Processor Messages 14 ISH to Host (ISH2HOST_MSG14)—Offset 1094h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.64 Outbound Inter Processor Messages 15 ISH to Host (ISH2HOST_MSG15)—Offset 1098h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.65 Outbound Inter Processor Messages 16 ISH to Host (ISH2HOST_MSG16)—Offset 109Ch

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.66 Outbound Inter Processor Messages 17 ISH to Host (ISH2HOST_MSG17)—Offset 10A0h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.67 Outbound Inter Processor Messages 18 ISH to Host (ISH2HOST_MSG18)—Offset 10A4h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.68 Outbound Inter Processor Messages 19 ISH to Host (ISH2HOST_MSG19)—Offset 10A8h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.69 Outbound Inter Processor Messages 20 ISH to Host (ISH2HOST_MSG20)—Offset 10ACh

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.70 Outbound Inter Processor Messages 21 ISH to Host (ISH2HOST_MSG21)—Offset 10B0h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.71 Outbound Inter Processor Messages 22 ISH to Host (ISH2HOST_MSG22)—Offset 10B4h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.72 Outbound Inter Processor Messages 23 ISH to Host (ISH2HOST_MSG23)—Offset 10B8h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.73 Outbound Inter Processor Messages 24 ISH to Host (ISH2HOST_MSG24)—Offset 10BCh

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.74 Outbound Inter Processor Messages 25 ISH to Host (ISH2HOST_MSG25)—Offset 10C0h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.75 Outbound Inter Processor Messages 26 ISH to Host (ISH2HOST_MSG26)—Offset 10C4h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.76 Outbound Inter Processor Messages 27 ISH to Host (ISH2HOST_MSG27)—Offset 10C8h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.77 Outbound Inter Processor Messages 28 ISH to Host (ISH2HOST_MSG28)—Offset 10CCh

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.78 Outbound Inter Processor Messages 29 ISH to Host (ISH2HOST_MSG29)—Offset 10D0h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.79 Outbound Inter Processor Messages 30 ISH to Host (ISH2HOST_MSG30)—Offset 10D4h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.80 Outbound Inter Processor Messages 31 ISH to Host (ISH2HOST_MSG31)—Offset 10D8h

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.81 Outbound Inter Processor Messages 32 ISH to Host (ISH2HOST_MSG32)—Offset 10DCh

Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from ISH to Host (MSG):** Inter-processor message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.82 Inbound Inter Processor Messages 1 Host to ISH (HOST2ISH_MSG1)—Offset 10E0h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.83 Inbound Inter Processor Messages 2 Host to ISH (HOST2ISH_MSG2)—Offset 10E4h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.84 Inbound Inter Processor Messages 3 Host to ISH (HOST2ISH_MSG3)—Offset 10E8h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.85 Inbound Inter Processor Messages 4 Host to ISH (HOST2ISH_MSG4)—Offset 10ECh

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.86 Inbound Inter Processor Messages 5 Host to ISH (HOST2ISH_MSG5)—Offset 10F0h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.87 Inbound Inter Processor Messages 6 Host to ISH (HOST2ISH_MSG6)—Offset 10F4h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.88 Inbound Inter Processor Messages 7 Host to ISH (HOST2ISH_MSG7)—Offset 10F8h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.89 Inbound Inter Processor Messages 8 Host to ISH (HOST2ISH_MSG8)—Offset 10FCh

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.90 Inbound Inter Processor Messages 9 Host to ISH (HOST2ISH_MSG9)—Offset 1100h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.91 Inbound Inter Processor Messages 10 Host to ISH (HOST2ISH_MSG10)—Offset 1104h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.92 Inbound Inter Processor Messages 11 Host to ISH (HOST2ISH_MSG11)—Offset 1108h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.93 Inbound Inter Processor Messages 12 Host to ISH (HOST2ISH_MSG12)—Offset 110Ch

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.94 Inbound Inter Processor Messages 13 Host to ISH (HOST2ISH_MSG13)—Offset 1110h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.95 Inbound Inter Processor Messages 14 Host to ISH (HOST2ISH_MSG14)—Offset 1114h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.96 Inbound Inter Processor Messages 15 Host to ISH (HOST2ISH_MSG15)—Offset 1118h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.97 Inbound Inter Processor Messages 16 Host to ISH (HOST2ISH_MSG16)—Offset 111Ch

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.98 Inbound Inter Processor Messages 17 Host to ISH (HOST2ISH_MSG17)—Offset 1120h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.99 Inbound Inter Processor Messages 18 Host to ISH (HOST2ISH_MSG18)—Offset 1124h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.100 Inbound Inter Processor Messages 19 Host to ISH (HOST2ISH_MSG19)—Offset 1128h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.101 Inbound Inter Processor Messages 20 Host to ISH (HOST2ISH_MSG20)—Offset 112Ch

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.102 Inbound Inter Processor Messages 21 Host to ISH (HOST2ISH_MSG21)—Offset 1130h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.103 Inbound Inter Processor Messages 22 Host to ISH (HOST2ISH_MSG22)—Offset 1134h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.104 Inbound Inter Processor Messages 23 Host to ISH (HOST2ISH_MSG23)—Offset 1138h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.105 Inbound Inter Processor Messages 24 Host to ISH (HOST2ISH_MSG24)—Offset 113Ch

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.106 Inbound Inter Processor Messages 25 Host to ISH (HOST2ISH_MSG25)—Offset 1140h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.107 Inbound Inter Processor Messages 26 Host to ISH (HOST2ISH_MSG26)—Offset 1144h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.108 Inbound Inter Processor Messages 27 Host to ISH (HOST2ISH_MSG27)—Offset 1148h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.109 Inbound Inter Processor Messages 28 Host to ISH (HOST2ISH_MSG28)—Offset 114Ch

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.110 Inbound Inter Processor Messages 29 Host to ISH (HOST2ISH_MSG29)—Offset 1150h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.111 Inbound Inter Processor Messages 30 Host to ISH (HOST2ISH_MSG30)—Offset 1154h

Inter-processor message registers for Host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from Host to ISH (MSG):** Inter-processor message registers for Host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.112 Inbound Inter Processor Messages 31 Host to ISH (HOST2ISH_MSG31)—Offset 1158h

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.113 Inbound Inter Processor Messages 32 Host to ISH (HOST2ISH_MSG32)—Offset 115Ch

Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from Host to ISH (MSG):** Inter-processor message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.114 Outbound Inter Processor Messages 1 ISH to SEC (ISH2SEC_MSG1)—Offset 1160h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.115 Outbound Inter Processor Messages 2 ISH to SEC (ISH2SEC_MSG2)—Offset 1164h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.116 Outbound Inter Processor Messages 3 ISH to SEC (ISH2SEC_MSG3)—Offset 1168h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.117 Outbound Inter Processor Messages 4 ISH to SEC (ISH2SEC_MSG4)—Offset 116Ch

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.118 Outbound Inter Processor Messages 5 ISH to SEC (ISH2SEC_MSG5)—Offset 1170h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.119 Outbound Inter Processor Messages 6 ISH to SEC (ISH2SEC_MSG6)—Offset 1174h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.120 Outbound Inter Processor Messages 7 ISH to SEC (ISH2SEC_MSG7)—Offset 1178h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.121 Outbound Inter Processor Messages 8 ISH to SEC (ISH2SEC_MSG8)—Offset 117Ch

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.122 Outbound Inter Processor Messages 9 ISH to SEC (ISH2SEC_MSG9)—Offset 1180h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.123 Outbound Inter Processor Messages 10 ISH to SEC (ISH2SEC_MSG10)—Offset 1184h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.124 Outbound Inter Processor Messages 11 ISH to SEC (ISH2SEC_MSG11)—Offset 1188h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.125 Outbound Inter Processor Messages 12 ISH to SEC (ISH2SEC_MSG12)—Offset 118Ch

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.126 Outbound Inter Processor Messages 13 ISH to SEC (ISH2SEC_MSG13)—Offset 1190h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.127 Outbound Inter Processor Messages 14 ISH to SEC (ISH2SEC_MSG14)—Offset 1194h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.128 Outbound Inter Processor Messages 15 ISH to SEC (ISH2SEC_MSG15)—Offset 1198h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.129 Outbound Inter Processor Messages 16 ISH to SEC (ISH2SEC_MSG16)—Offset 119Ch

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.130 Outbound Inter Processor Messages 17 ISH to SEC (ISH2SEC_MSG17)—Offset 11A0h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.131 Outbound Inter Processor Messages 18 ISH to SEC (ISH2SEC_MSG18)—Offset 11A4h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.132 Outbound Inter Processor Messages 19 ISH to SEC (ISH2SEC_MSG19)—Offset 11A8h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.133 Outbound Inter Processor Messages 20 ISH to SEC (ISH2SEC_MSG20)—Offset 11ACh

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.134 Outbound Inter Processor Messages 21 ISH to SEC (ISH2SEC_MSG21)—Offset 11B0h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.135 Outbound Inter Processor Messages 22 ISH to SEC (ISH2SEC_MSG22)—Offset 11B4h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.136 Outbound Inter Processor Messages 23 ISH to SEC (ISH2SEC_MSG23)—Offset 11B8h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.137 Outbound Inter Processor Messages 24 ISH to SEC (ISH2SEC_MSG24)—Offset 11BCh

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.138 Outbound Inter Processor Messages 25 ISH to SEC (ISH2SEC_MSG25)—Offset 11C0h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.139 Outbound Inter Processor Messages 26 ISH to SEC (ISH2SEC_MSG26)—Offset 11C4h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.140 Outbound Inter Processor Messages 27 ISH to SEC (ISH2SEC_MSG27)—Offset 11C8h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.141 Outbound Inter Processor Messages 28 ISH to SEC (ISH2SEC_MSG28)—Offset 11CCh

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.142 Outbound Inter Processor Messages 29 ISH to SEC (ISH2SEC_MSG29)—Offset 11D0h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.143 Outbound Inter Processor Messages 30 ISH to SEC (ISH2SEC_MSG30)—Offset 11D4h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.144 Outbound Inter Processor Messages 31 ISH to SEC (ISH2SEC_MSG31)—Offset 11D8h

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.145 Outbound Inter Processor Messages 32 ISH to SEC (ISH2SEC_MSG32)—Offset 11DCh

Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from ISH to SEC (MSG):** Inter-processor message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.146 Inbound Inter Processor Messages 1 SEC to ISH (SEC2ISH_MSG1)—Offset 11E0h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.147 Inbound Inter Processor Messages 2 SEC to ISH (SEC2ISH_MSG2)—Offset 11E4h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.148 Inbound Inter Processor Messages 3 SEC to ISH (SEC2ISH_MSG3)—Offset 11E8h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.149 Inbound Inter Processor Messages 4 SEC to ISH (SEC2ISH_MSG4)—Offset 11ECh

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.150 Inbound Inter Processor Messages 5 SEC to ISH (SEC2ISH_MSG5)—Offset 11F0h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.151 Inbound Inter Processor Messages 6 SEC to ISH (SEC2ISH_MSG6)—Offset 11F4h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.152 Inbound Inter Processor Messages 7 SEC to ISH (SEC2ISH_MSG7)—Offset 11F8h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.153 Inbound Inter Processor Messages 8 SEC to ISH (SEC2ISH_MSG8)—Offset 11FCh

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.154 Inbound Inter Processor Messages 9 SEC To ISH (SEC2ISH_MSG9)—Offset 1200h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.155 Inbound Inter Processor Messages 10 SEC to ISH (SEC2ISH_MSG10)—Offset 1204h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.156 Inbound Inter Processor Messages 11 SEC to ISH (SEC2ISH_MSG11)—Offset 1208h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.157 Inbound Inter Processor Messages 12 SEC to ISH (SEC2ISH_MSG12)—Offset 120Ch

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.158 Inbound Inter Processor Messages 13 SEC to ISH (SEC2ISH_MSG13)—Offset 1210h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.159 Inbound Inter Processor Messages 14 SEC to ISH (SEC2ISH_MSG14)—Offset 1214h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.160 Inbound Inter Processor Messages 15 SEC to ISH (SEC2ISH_MSG15)—Offset 1218h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.161 Inbound Inter Processor Messages 16 SEC to ISH (SEC2ISH_MSG16)—Offset 121Ch

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.162 Inbound Inter Processor Messages 17 SEC to ISH (SEC2ISH_MSG17)—Offset 1220h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.163 Inbound Inter Processor Messages 18 SEC to ISH (SEC2ISH_MSG18)—Offset 1224h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.164 Inbound Inter Processor Messages 19 SEC to ISH (SEC2ISH_MSG19)—Offset 1228h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.165 Inbound Inter Processor Messages 20 SEC to ISH (SEC2ISH_MSG20)—Offset 122Ch

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.166 Inbound Inter Processor Messages 21 SEC to ISH (SEC2ISH_MSG21)—Offset 1230h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.167 Inbound Inter Processor Messages 22 SEC to ISH (SEC2ISH_MSG22)—Offset 1234h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.168 Inbound Inter Processor Messages 23 SEC to ISH (SEC2ISH_MSG23)—Offset 1238h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.169 Inbound Inter Processor Messages 24 SEC to ISH (SEC2ISH_MSG24)—Offset 123Ch

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.170 Inbound Inter Processor Messages 25 SEC to ISH (SEC2ISH_MSG25)—Offset 1240h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.171 Inbound Inter Processor Messages 26 SEC to ISH (SEC2ISH_MSG26)—Offset 1244h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.172 Inbound Inter Processor Messages 27 SEC to ISH (SEC2ISH_MSG27)—Offset 1248h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.173 Inbound Inter Processor Messages 28 SEC to ISH (SEC2ISH_MSG28)—Offset 124Ch

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.174 Inbound Inter Processor Messages 29 SEC To ISH (SEC2ISH_MSG29)—Offset 1250h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.175 Inbound Inter Processor Messages 30 SEC To ISH (SEC2ISH_MSG30)—Offset 1254h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.176 Inbound Inter Processor Messages 31 SEC To ISH (SEC2ISH_MSG31)—Offset 1258h

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.177 Inbound Inter Processor Messages 32 SEC to ISH (SEC2ISH_MSG32)—Offset 125Ch

Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from SEC to ISH (MSG):** Inter-processor message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.178 Outbound Inter Processor Messages 1 ISH to PMC (ISH2PMC_MSG1)—Offset 1260h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.179 Outbound Inter Processor Messages 2 ISH to PMC (ISH2PMC_MSG2)—Offset 1264h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.180 Outbound Inter Processor Messages 3 ISH to PMC (ISH2PMC_MSG3)—Offset 1268h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.181 Outbound Inter Processor Messages 4 ISH to PMC (ISH2PMC_MSG4)—Offset 126Ch

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.182 Outbound Inter Processor Messages 5 ISH to PMC (ISH2PMC_MSG5)—Offset 1270h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.183 Outbound Inter Processor Messages 6 ISH to PMC (ISH2PMC_MSG6)—Offset 1274h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.184 Outbound Inter Processor Messages 7 ISH to PMC (ISH2PMC_MSG7)—Offset 1278h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.185 Outbound Inter Processor Messages 8 ISH to PMC (ISH2PMC_MSG8)—Offset 127Ch

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.186 Outbound Inter Processor Messages 9 ISH to PMC (ISH2PMC_MSG9)—Offset 1280h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.187 Outbound Inter Processor Messages 10 ISH to PMC (ISH2PMC_MSG10)—Offset 1284h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.188 Outbound Inter Processor Messages 11 ISH to PMC (ISH2PMC_MSG11)—Offset 1288h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.189 Outbound Inter Processor Messages 12 ISH to PMC (ISH2PMC_MSG12)—Offset 128Ch

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.190 Outbound Inter Processor Messages 13 ISH to PMC (ISH2PMC_MSG13)—Offset 1290h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.191 Outbound Inter Processor Messages 14 ISH to PMC (ISH2PMC_MSG14)—Offset 1294h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.192 Outbound Inter Processor Messages 15 ISH to PMC (ISH2PMC_MSG15)—Offset 1298h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.193 Outbound Inter Processor Messages 16 ISH to PMC (ISH2PMC_MSG16)—Offset 129Ch

Inter-processor Message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from ISH to PMC (MSG):** Inter-processor Message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.194 Outbound Inter Processor Messages 17 ISH to PMC (ISH2PMC_MSG17)—Offset 12A0h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.195 Outbound Inter Processor Messages 18 ISH to PMC (ISH2PMC_MSG18)—Offset 12A4h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.196 Outbound Inter Processor Messages 19 ISH to PMC (ISH2PMC_MSG19)—Offset 12A8h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.197 Outbound Inter Processor Messages 20 ISH To PMC (ISH2PMC_MSG20)—Offset 12ACh

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.198 Outbound Inter Processor Messages 21 ISH to PMC (ISH2PMC_MSG21)—Offset 12B0h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.199 Outbound Inter Processor Messages 22 ISH to PMC (ISH2PMC_MSG22)—Offset 12B4h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.200 Outbound Inter Processor Messages 23 ISH to PMC (ISH2PMC_MSG23)—Offset 12B8h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.201 Outbound Inter Processor Messages 24 ISH to PMC (ISH2PMC_MSG24)—Offset 12BCh

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.202 Outbound Inter Processor Messages 25 ISH to PMC (ISH2PMC_MSG25)—Offset 12C0h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.203 Outbound Inter Processor Messages 26 ISH to PMC (ISH2PMC_MSG26)—Offset 12C4h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.204 Outbound Inter Processor Messages 27 ISH to PMC (ISH2PMC_MSG27)—Offset 12C8h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.205 Outbound Inter Processor Messages 28 ISH to PMC (ISH2PMC_MSG28)—Offset 12CCh

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.206 Outbound Inter Processor Messages 29 ISH to PMC (ISH2PMC_MSG29)—Offset 12D0h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.207 Outbound Inter Processor Messages 30 ISH to PMC (ISH2PMC_MSG30)—Offset 12D4h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.208 Outbound Inter Processor Messages 31 ISH to PMC (ISH2PMC_MSG31)—Offset 12D8h

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.209 Outbound Inter Processor Messages 32 ISH to PMC (ISH2PMC_MSG32)—Offset 12DCh

Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from ISH to PMC (MSG):** Inter-processor message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.210 Inbound Inter Processor Messages 1 PMC to ISH (PMC2ISH_MSG1)—Offset 12E0h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.211 Inbound Inter Processor Messages 2 PMC to ISH (PMC2ISH_MSG2)—Offset 12E4h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.212 Inbound Inter Processor Messages 3 PMC to ISH (PMC2ISH_MSG3)—Offset 12E8h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.213 Inbound Inter Processor Messages 4 PMC to ISH (PMC2ISH_MSG4)—Offset 12ECh

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.214 Inbound Inter Processor Messages 5 PMC to ISH (PMC2ISH_MSG5)—Offset 12F0h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.215 Inbound Inter Processor Messages 6 PMC to ISH (PMC2ISH_MSG6)—Offset 12F4h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.216 Inbound Inter Processor Messages 7 PMC to ISH (PMC2ISH_MSG7)—Offset 12F8h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.217 Inbound Inter Processor Messages 8 PMC to ISH (PMC2ISH_MSG8)—Offset 12FCh

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.218 Inbound Inter Processor Messages 9 PMC to ISH (PMC2ISH_MSG9)—Offset 1300h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.219 Inbound Inter Processor Messages 10 PMC to ISH (PMC2ISH_MSG10)—Offset 1304h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.220 Inbound Inter Processor Messages 11 PMC to ISH (PMC2ISH_MSG11)—Offset 1308h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.221 Inbound Inter Processor Messages 12 PMC to ISH (PMC2ISH_MSG12)—Offset 130Ch

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.222 Inbound Inter Processor Messages 13 PMC to ISH (PMC2ISH_MSG13)—Offset 1310h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.223 Inbound Inter Processor Messages 14 PMC to ISH (PMC2ISH_MSG14)—Offset 1314h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.224 Inbound Inter Processor Messages 15 PMC to ISH (PMC2ISH_MSG15)—Offset 1318h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.225 Inbound Inter Processor Messages 16 PMC to ISH (PMC2ISH_MSG16)—Offset 131Ch

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.226 Inbound Inter Processor Messages 17 PMC to ISH (PMC2ISH_MSG17)—Offset 1320h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.227 Inbound Inter Processor Messages 18 PMC to ISH (PMC2ISH_MSG18)—Offset 1324h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.228 Inbound Inter Processor Messages 19 PMC to ISH (PMC2ISH_MSG19)—Offset 1328h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.229 Inbound Inter Processor Messages 20 PMC to ISH (PMC2ISH_MSG20)—Offset 132Ch

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.230 Inbound Inter Processor Messages 21 PMC to ISH (PMC2ISH_MSG21)—Offset 1330h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.231 Inbound Inter Processor Messages 22 PMC to ISH (PMC2ISH_MSG22)—Offset 1334h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.232 Inbound Inter Processor Messages 23 PMC to ISH (PMC2ISH_MSG23)—Offset 1338h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.233 Inbound Inter Processor Messages 24 PMC to ISH (PMC2ISH_MSG24)—Offset 133Ch

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.234 Inbound Inter Processor Messages 25 PMC to ISH (PMC2ISH_MSG25)—Offset 1340h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.235 Inbound Inter Processor Messages 26 PMC to ISH (PMC2ISH_MSG26)—Offset 1344h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.236 Inbound Inter Processor Messages 27 PMC to ISH (PMC2ISH_MSG27)—Offset 1348h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.237 Inbound Inter Processor Messages 28 PMC to ISH (PMC2ISH_MSG28)—Offset 134Ch

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.238 Inbound Inter Processor Messages 29 PMC to ISH (PMC2ISH_MSG29)—Offset 1350h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.239 Inbound Inter Processor Messages 30 PMC to ISH (PMC2ISH_MSG30)—Offset 1354h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.240 Inbound Inter Processor Messages 31 PMC to ISH (PMC2ISH_MSG31)—Offset 1358h

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.241 Inbound Inter Processor Messages 32 PMC to ISH (PMC2ISH_MSG32)—Offset 135Ch

Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from PMC to ISH (MSG):** Inter-processor message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.242 Remap 0 (REMAP0)—Offset 1360h

At boot time, the host can write these registers with any dynamically allocated address spaces that ISH will have to access.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Remap 0 (REMAP0)**

20.243 Remap 1 (REMAP1)—Offset 1364h

At boot time, the host can write these registers with any dynamically allocated address spaces that ISH will have to access.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Remap 1 (REMAP1)**

20.244 Remap 2 (REMAP2)—Offset 1368h

At boot time, the host can write these registers with any dynamically allocated address spaces that ISH will have to access.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Remap 2 (REMAP2)**

20.245 Remap 3 (REMAP3)—Offset 136Ch

At boot time, the host can write these registers with any dynamically allocated address spaces that ISH will have to access.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Remap 3 (REMAP3)**

20.246 Remap 4 (REMAP4)—Offset 1370h

At boot time, the host can write these registers with any dynamically allocated address spaces that ISH will have to access.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Remap 4 (REMAP4)**

20.247 Remap 5 (REMAP5)—Offset 1374h

At boot time, the host can write these registers with any dynamically allocated address spaces that ISH will have to access.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Remap 5 (REMAP5)**

20.248 ISH IPC Busy Clear (ISH_IPC_BUSY_CLEAR)—Offset 1378h

This register holds the status of the ISH IPC busy clear interrupts. ISH IPC busy clear interrupt is set when busy bit of respective outbound doorbell register gets cleared and interrupt is cleared when ISH writes CAN1EM to respective bit in ISH IPC busy clear register.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RESERVED0):** Reserved.
16	0h RW/1C	**ISH to SEC Busy Clear Status (ISH2SEC_BUSY_CLEAR):** Busy clear interrupt bit of ISH to SEC IPC: • 1: Interrupt active. • 0: Interrupt inactive.
15:9	0h RO	**Reserved (RESERVED1):** Reserved.
8	0h RW/1C	**ISH to PMC Busy Clear Status (ISH2PMC_BUSY_CLEAR):** Busy clear interrupt bit of ISH to PMC IPC: • 1: Interrupt active. • 0: Interrupt inactive.
7:4	0h RO	**Reserved (RESERVED2):** Reserved.
3	0h RW/1C	**ISH to Audio Busy Clear Status (ISH2AUDIO_BUSY_CLEAR):** Busy clear interrupt bit of ISH to Audio IPC: • 1: Interrupt active. • 0: Interrupt inactive.
2	0h RW/1C	**ISH to ISP Busy Clear Status (ISH2ISP_BUSY_CLEAR):** Busy clear interrupt bit of ISH to ISP IPC: • 1: Interrupt active. • 0: Interrupt inactive.
1	0h RW/1C	**ISH to GFX Busy Clear Status (ISH2GFX_BUSY_CLEAR):** Busy clear interrupt bit of ISH to GFX IPC: • 1: Interrupt active. • 0: Interrupt inactive.
0	0h RW/1C	**ISH to Host Busy Clear Status (ISH2HOST_BUSY_CLEAR):** Busy clear interrupt bit of ISH to host IPC: • 1: Interrupt active. • 0: Interrupt inactive.

20.249 ISH Fuse (ISH_FUSE)—Offset 137Ch

This register mirrors the ISH fuse bits.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RESERVED):** Reserved.
3:0	0h RO	**ISH Fuse (ISH_FUSE)**

20.250 UMA Base Address LSB (UMA_RANGE_LOWER_0)—Offset 1380h

This register should be accessed only in ISH 3.0. This is an address range register, holds 32 bit LSB value of the address for UMA base range.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: FFFFF001h

Bit Range	Default & Access	Field Name (ID): Description
31:12	FFFFFh RW	**UMA Space Lower Memory Base Address (UMA_MEMORY_BASE_ADDR_LSB)**
11:1	0h RO	**Size (SIZE):** Memory size, hard wired to 0 indicate 4 KB window granularity.
0	1h RW	**Memory Range Disabled (SIZE_BIT0):** • 1 means No TC7 or RS = 1 cycles, • 0 means TC7 or RS =1.

20.251 UMA Base Address MSB (UMA_RANGE_LOWER_1)—Offset 1384h

This register should be accessed only in ISH 3.0. This is an address range register, holds 32 bit MSB value of the address for UMA base range.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**UMA Memory Base Address MSB (UMA_MEMORY_BASE_ADDR_MSB)**

20.252 UMA Limit Address LSB (UMA_RANGE_UPPER_0)—Offset 1388h

This register should be accessed only in ISH3.0. This is an address range register, holds 32 bit LSB value of the address for UMA limit range.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: FFFFF000h

Bit Range	Default & Access	Field Name (ID): Description
31:12	FFFFFh RW	**UMA Memory Limit Address LSB (UMA_MEMORY_LIMIT_ADDR_LSB)**
11:0	0h RO	**Size (SIZE):** UMA memory size, hard wired to 0 indicate 4 KB window granularity.

20.253 UMA Base Address MSB (UMA_RANGE_UPPER_1)—Offset 138Ch

This register should be accessed only in ISH 3.0. This is an address range register, holds 32 bit MSB value of the address for UMA base range.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**UMA Memory Limit Address MSB (UMA_MEMORY_LIMIT_ADDR_MSB)**

20.254 Outbound Inter Processor Messages 1 ISH to ISP (ISH2ISP_MSG1)—Offset 1390h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.255 Outbound Inter Processor Messages 2 ISH to ISP (ISH2ISP_MSG2)—Offset 1394h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.256 Outbound Inter Processor Messages 3 ISH to ISP (ISH2ISP_MSG3)—Offset 1398h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.257 Outbound Inter Processor Messages 4 ISH to ISP (ISH2ISP_MSG4)—Offset 139Ch

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.258 Outbound Inter Processor Messages 5 ISH to ISP (ISH2ISP_MSG5)—Offset 13A0h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.259 Outbound Inter Processor Messages 6 ISH to ISP (ISH2ISP_MSG6)—Offset 13A4h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.260 Outbound Inter Processor Messages 7 ISH to ISP (ISH2ISP_MSG7)—Offset 13A8h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.261 Outbound Inter Processor Messages 8 ISH to ISP (ISH2ISP_MSG8)—Offset 13ACh

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.262 Outbound Inter Processor Messages 9 ISH to ISP (ISH2ISP_MSG9)—Offset 13B0h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.263 Outbound Inter Processor Messages 10 ISH to ISP (ISH2ISP_MSG10)—Offset 13B4h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.264 Outbound Inter Processor Messages 11 ISH to ISP (ISH2ISP_MSG11)—Offset 13B8h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.265 Outbound Inter Processor Messages 12 ISH to ISP (ISH2ISP_MSG12)—Offset 13BCh

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.266 Outbound Inter Processor Messages 13 ISH to ISP (ISH2ISP_MSG13)—Offset 13C0h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.267 Outbound Inter Processor Messages 14 ISH to ISP (ISH2ISP_MSG14)—Offset 13C4h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.268 Outbound Inter Processor Messages 15 ISH to ISP (ISH2ISP_MSG15)—Offset 13C8h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.269 Outbound Inter Processor Messages 16 ISH to ISP (ISH2ISP_MSG16)—Offset 13CCh

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.270 Outbound Inter Processor Messages 17 ISH to ISP (ISH2ISP_MSG17)—Offset 13D0h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.271 Outbound Inter Processor Messages 18 ISH to ISP (ISH2ISP_MSG18)—Offset 13D4h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.272 Outbound Inter Processor Messages 19 ISH to ISP (ISH2ISP_MSG19)—Offset 13D8h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.273 Outbound Inter Processor Messages 20 ISH to ISP (ISH2ISP_MSG20)—Offset 13DCh

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.274 Outbound Inter Processor Messages 21 ISH To ISP (ISH2ISP_MSG21)—Offset 13E0h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.275 Outbound Inter Processor Messages 22 ISH to ISP (ISH2ISP_MSG22)—Offset 13E4h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.276 Outbound Inter Processor Messages 23 ISH to ISP (ISH2ISP_MSG23)—Offset 13E8h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.277 Outbound Inter Processor Messages 24 ISH to ISP (ISH2ISP_MSG24)—Offset 13ECh

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.278 Outbound Inter Processor Messages 25 ISH to ISP (ISH2ISP_MSG25)—Offset 13F0h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.279 Outbound Inter Processor Messages 26 ISH to ISP (ISH2ISP_MSG26)—Offset 13F4h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.280 Outbound Inter Processor Messages 27 ISH to ISP (ISH2ISP_MSG27)—Offset 13F8h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.281 Outbound Inter Processor Messages 28 ISH to ISP (ISH2ISP_MSG28)—Offset 13FCh

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.282 Outbound Inter Processor Messages 29 ISH to ISP (ISH2ISP_MSG29)—Offset 1400h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.283 Outbound Inter Processor Messages 30 ISH to ISP (ISH2ISP_MSG30)—Offset 1404h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.284 Outbound Inter Processor Messages 31 ISH to ISP (ISH2ISP_MSG31)—Offset 1408h

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.285 Outbound Inter Processor Messages 32 ISH to ISP (ISH2ISP_MSG32)—Offset 140Ch

Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from ISH to ISP (MSG):** Inter-processor message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.286 Inbound Inter Processor Messages 1 ISP to ISH (ISP2ISH_MSG1)—Offset 1410h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.287 Inbound Inter Processor Messages 2 ISP to ISH (ISP2ISH_MSG2)—Offset 1414h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.288 Inbound Inter Processor Messages 3 ISP to ISH (ISP2ISH_MSG3)—Offset 1418h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.289 Inbound Inter Processor Messages 4 ISP to ISH (ISP2ISH_MSG4)—Offset 141Ch

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.290 Inbound Inter Processor Messages 5 ISP to ISH (ISP2ISH_MSG5)—Offset 1420h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.291 Inbound Inter Processor Messages 6 ISP to ISH (ISP2ISH_MSG6)—Offset 1424h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.292 Inbound Inter Processor Messages 7 ISP to ISH (ISP2ISH_MSG7)—Offset 1428h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.293 Inbound Inter Processor Messages 8 ISP to ISH (ISP2ISH_MSG8)—Offset 142Ch

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.294 Inbound Inter Processor Messages 9 ISP to ISH (ISP2ISH_MSG9)—Offset 1430h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.295 Inbound Inter Processor Messages 10 ISP to ISH (ISP2ISH_MSG10)—Offset 1434h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.296 Inbound Inter Processor Messages 11 ISP to ISH (ISP2ISH_MSG11)—Offset 1438h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.297 Inbound Inter Processor Messages 12 ISP to ISH (ISP2ISH_MSG12)—Offset 143Ch

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.298 Inbound Inter Processor Messages 13 ISP to ISH (ISP2ISH_MSG13)—Offset 1440h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.299 Inbound Inter Processor Messages 14 ISP to ISH (ISP2ISH_MSG14)—Offset 1444h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.300 Inbound Inter Processor Messages 15 ISP to ISH (ISP2ISH_MSG15)—Offset 1448h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.301 Inbound Inter Processor Messages 16 ISP to ISH (ISP2ISH_MSG16)—Offset 144Ch

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.302 Inbound Inter Processor Messages 17 ISP to ISH (ISP2ISH_MSG17)—Offset 1450h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.303 Inbound Inter Processor Messages 18 ISP to ISH (ISP2ISH_MSG18)—Offset 1454h

Inter-processor Message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from ISP to ISH (MSG):** Inter-processor Message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.304 Inbound Inter Processor Messages 19 ISP to ISH (ISP2ISH_MSG19)—Offset 1458h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.305 Inbound Inter Processor Messages 20 ISP to ISH (ISP2ISH_MSG20)—Offset 145Ch

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.306 Inbound Inter Processor Messages 21 ISP to ISH (ISP2ISH_MSG21)—Offset 1460h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.307 Inbound Inter Processor Messages 22 ISP to ISH (ISP2ISH_MSG22)—Offset 1464h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.308 Inbound Inter Processor Messages 23 ISP to ISH (ISP2ISH_MSG23)—Offset 1468h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.309 Inbound Inter Processor Messages 24 ISP to ISH (ISP2ISH_MSG24)—Offset 146Ch

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.310 Inbound Inter Processor Messages 25 ISP to ISH (ISP2ISH_MSG25)—Offset 1470h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.311 Inbound Inter Processor Messages 26 ISP to ISH (ISP2ISH_MSG26)—Offset 1474h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.312 Inbound Inter Processor Messages 27 ISP to ISH (ISP2ISH_MSG27)—Offset 1478h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.313 Inbound Inter Processor Messages 28 ISP to ISH (ISP2ISH_MSG28)—Offset 147Ch

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.314 Inbound Inter Processor Messages 29 ISP to ISH (ISP2ISH_MSG29)—Offset 1480h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.315 Inbound Inter Processor Messages 30 ISP to ISH (ISP2ISH_MSG30)—Offset 1484h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.316 Inbound Inter Processor Messages 31 ISP to ISH (ISP2ISH_MSG31)—Offset 1488h

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.317 Inbound Inter Processor Messages 32 ISP to ISH (ISP2ISH_MSG32)—Offset 148Ch

Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from ISP to ISH (MSG):** Inter-processor message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.318 Outbound Inter Processor Messages 1 ISH to Audio (ISH2AUDIO_MSG1)—Offset 1490h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.319 Outbound Inter Processor Messages 2 ISH to Audio (ISH2AUDIO_MSG2)—Offset 1494h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.320 Outbound Inter Processor Messages 3 ISH to Audio (ISH2AUDIO_MSG3)—Offset 1498h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.321 Outbound Inter Processor Messages 4 ISH to Audio (ISH2AUDIO_MSG4)—Offset 149Ch

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.322 Outbound Inter Processor Messages 5 ISH to Audio (ISH2AUDIO_MSG5)—Offset 14A0h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.323 Outbound Inter Processor Messages 6 ISH to Audio (ISH2AUDIO_MSG6)—Offset 14A4h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.324 Outbound Inter Processor Messages 7 ISH to Audio (ISH2AUDIO_MSG7)—Offset 14A8h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.325 Outbound Inter Processor Messages 8 ISH to Audio (ISH2AUDIO_MSG8)—Offset 14ACh

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.326 Outbound Inter Processor Messages 9 ISH to Audio (ISH2AUDIO_MSG9)—Offset 14B0h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.327 Outbound Inter Processor Messages 10 ISH to Audio (ISH2AUDIO_MSG10)—Offset 14B4h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.328 Outbound Inter Processor Messages 11 ISH to Audio (ISH2AUDIO_MSG11)—Offset 14B8h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.329 Outbound Inter Processor Messages 12 ISH to Audio (ISH2AUDIO_MSG12)—Offset 14BCh

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.330 Outbound Inter Processor Messages 13 ISH to Audio (ISH2AUDIO_MSG13)—Offset 14C0h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.331 Outbound Inter Processor Messages 14 ISH to Audio (ISH2AUDIO_MSG14)—Offset 14C4h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.332 Outbound Inter Processor Messages 15 ISH to Audio (ISH2AUDIO_MSG15)—Offset 14C8h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.333 Outbound Inter Processor Messages 16 ISH to Audio (ISH2AUDIO_MSG16)—Offset 14CCh

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.334 Outbound Inter Processor Messages 17 ISH to Audio (ISH2AUDIO_MSG17)—Offset 14D0h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.335 Outbound Inter Processor Messages 18 ISH to Audio (ISH2AUDIO_MSG18)—Offset 14D4h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.336 Outbound Inter Processor Messages 19 ISH to Audio (ISH2AUDIO_MSG19)—Offset 14D8h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.337 Outbound Inter Processor Messages 20 ISH to Audio (ISH2AUDIO_MSG20)—Offset 14DCh

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.338 Outbound Inter Processor Messages 21 ISH to Audio (ISH2AUDIO_MSG21)—Offset 14E0h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.339 Outbound Inter Processor Messages 22 ISH to Audio (ISH2AUDIO_MSG22)—Offset 14E4h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.340 Outbound Inter Processor Messages 23 ISH to Audio (ISH2AUDIO_MSG23)—Offset 14E8h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.341 Outbound Inter Processor Messages 24 ISH to Audio (ISH2AUDIO_MSG24)—Offset 14ECh

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.342 Outbound Inter Processor Messages 25 ISH to Audio (ISH2AUDIO_MSG25)—Offset 14F0h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.343 Outbound Inter Processor Messages 26 ISH to Audio (ISH2AUDIO_MSG26)—Offset 14F4h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.344 Outbound Inter Processor Messages 27 ISH to Audio (ISH2AUDIO_MSG27)—Offset 14F8h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.345 Outbound Inter Processor Messages 28 ISH to Audio (ISH2AUDIO_MSG28)—Offset 14FCh

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.346 Outbound Inter Processor Messages 29 ISH to Audio (ISH2AUDIO_MSG29)—Offset 1500h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.347 Outbound Inter Processor Messages 30 ISH to Audio (ISH2AUDIO_MSG30)—Offset 1504h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.348 Outbound Inter Processor Messages 31 ISH to Audio (ISH2AUDIO_MSG31)—Offset 1508h

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.349 Outbound Inter Processor Messages 32 ISH to Audio (ISH2AUDIO_MSG32)—Offset 150Ch

Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from ISH to Audio (MSG):** Inter-processor message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.350 Inbound Inter Processor Messages 1 Audio to ISH (AUDIO2ISH_MSG1)—Offset 1510h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.351 Inbound Inter Processor Messages 2 Audio to ISH (AUDIO2ISH_MSG2)—Offset 1514h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.352 Inbound Inter Processor Messages 3 Audio to ISH (AUDIO2ISH_MSG3)—Offset 1518h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.353 Inbound Inter Processor Messages 4 Audio to ISH (AUDIO2ISH_MSG4)—Offset 151Ch

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.354 Inbound Inter Processor Messages 5 Audio to ISH (AUDIO2ISH_MSG5)—Offset 1520h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.355 Inbound Inter Processor Messages 6 Audio to ISH (AUDIO2ISH_MSG6)—Offset 1524h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.356 Inbound Inter Processor Messages 7 Audio to ISH (AUDIO2ISH_MSG7)—Offset 1528h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.357 Inbound Inter Processor Messages 8 Audio to ISH (AUDIO2ISH_MSG8)—Offset 152Ch

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.358 Inbound Inter Processor Messages 9 Audio to ISH (AUDIO2ISH_MSG9)—Offset 1530h

Inter-processor Message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from Audio to ISH (MSG):** Inter-processor Message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.359 Inbound Inter Processor Messages 10 Audio to ISH (AUDIO2ISH_MSG10)—Offset 1534h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.360 Inbound Inter Processor Messages 11 Audio to ISH (AUDIO2ISH_MSG11)—Offset 1538h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.361 Inbound Inter Processor Messages 12 Audio to ISH (AUDIO2ISH_MSG12)—Offset 153Ch

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.362 Inbound Inter Processor Messages 13 Audio to ISH (AUDIO2ISH_MSG13)—Offset 1540h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.363 Inbound Inter Processor Messages 14 Audio to ISH (AUDIO2ISH_MSG14)—Offset 1544h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.364 Inbound Inter Processor Messages 15 Audio to ISH (AUDIO2ISH_MSG15)—Offset 1548h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.365 Inbound Inter Processor Messages 16 Audio to ISH (AUDIO2ISH_MSG16)—Offset 154Ch

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.366 Inbound Inter Processor Messages 17 Audio to ISH (AUDIO2ISH_MSG17)—Offset 1550h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.367 Inbound Inter Processor Messages 18 Audio to ISH (AUDIO2ISH_MSG18)—Offset 1554h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.368 Inbound Inter Processor Messages 19 Audio to ISH (AUDIO2ISH_MSG19)—Offset 1558h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.369 Inbound Inter Processor Messages 20 Audio to ISH (AUDIO2ISH_MSG20)—Offset 155Ch

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.370 Inbound Inter Processor Messages 21 Audio to ISH (AUDIO2ISH_MSG21)—Offset 1560h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.371 Inbound Inter Processor Messages 22 Audio to ISH (AUDIO2ISH_MSG22)—Offset 1564h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.372 Inbound Inter Processor Messages 23 Audio to ISH (AUDIO2ISH_MSG23)—Offset 1568h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.373 Inbound Inter Processor Messages 24 Audio to ISH (AUDIO2ISH_MSG24)—Offset 156Ch

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.374 Inbound Inter Processor Messages 25 Audio to ISH (AUDIO2ISH_MSG25)—Offset 1570h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.375 Inbound Inter Processor Messages 26 Audio to ISH (AUDIO2ISH_MSG26)—Offset 1574h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.376 Inbound Inter Processor Messages 27 Audio to ISH (AUDIO2ISH_MSG27)—Offset 1578h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.377 Inbound Inter Processor Messages 28 Audio to ISH (AUDIO2ISH_MSG28)—Offset 157Ch

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.378 Inbound Inter Processor Messages 29 Audio to ISH (AUDIO2ISH_MSG29)—Offset 1580h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.379 Inbound Inter Processor Messages 30 Audio to ISH (AUDIO2ISH_MSG30)—Offset 1584h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.380 Inbound Inter Processor Messages 31 Audio to ISH (AUDIO2ISH_MSG31)—Offset 1588h

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.381 Inbound Inter Processor Messages 32 Audio to ISH (AUDIO2ISH_MSG32)—Offset 158Ch

Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from Audio to ISH (MSG):** Inter-processor message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.382 Outbound Inter Processor Messages 1 ISH to Gfx (ISH2GFX_MSG1)—Offset 1590h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.383 Outbound Inter Processor Messages 2 ISH to Gfx (ISH2GFX_MSG2)—Offset 1594h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.384 Outbound Inter Processor Messages 3 ISH to Gfx (ISH2GFX_MSG3)—Offset 1598h

Inter-processor Message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from ISH to GFX (MSG):** Inter-processor Message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.385 Outbound Inter Processor Messages 4 ISH to Gfx (ISH2GFX_MSG4)—Offset 159Ch

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.386 Outbound Inter Processor Messages 5 ISH to Gfx (ISH2GFX_MSG5)—Offset 15A0h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.387 Outbound Inter Processor Messages 6 ISH to Gfx (ISH2GFX_MSG6)—Offset 15A4h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.388 Outbound Inter Processor Messages 7 ISH to Gfx (ISH2GFX_MSG7)—Offset 15A8h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.389 Outbound Inter Processor Messages 8 ISH to Gfx (ISH2GFX_MSG8)—Offset 15ACh

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.390 Outbound Inter Processor Messages 9 ISH to Gfx (ISH2GFX_MSG9)—Offset 15B0h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.391 Outbound Inter Processor Messages 10 ISH to Gfx (ISH2GFX_MSG10)—Offset 15B4h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.392 Outbound Inter Processor Messages 11 ISH to Gfx (ISH2GFX_MSG11)—Offset 15B8h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.393 Outbound Inter Processor Messages 12 ISH to Gfx (ISH2GFX_MSG12)—Offset 15BCh

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.394 Outbound Inter Processor Messages 13 ISH to Gfx (ISH2GFX_MSG13)—Offset 15C0h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.395 Outbound Inter Processor Messages 14 ISH to Gfx (ISH2GFX_MSG14)—Offset 15C4h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.396 Outbound Inter Processor Messages 15 ISH to Gfx (ISH2GFX_MSG15)—Offset 15C8h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.397 Outbound Inter Processor Messages 16 ISH to Gfx (ISH2GFX_MSG16)—Offset 15CCh

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.398 Outbound Inter Processor Messages 17 ISH to Gfx (ISH2GFX_MSG17)—Offset 15D0h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.399 Outbound Inter Processor Messages 18 ISH to Gfx (ISH2GFX_MSG18)—Offset 15D4h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.400 Outbound Inter Processor Messages 19 ISH to Gfx (ISH2GFX_MSG19)—Offset 15D8h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.401 Outbound Inter Processor Messages 20 ISH to Gfx (ISH2GFX_MSG20)—Offset 15DCh

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.402 Outbound Inter Processor Messages 21 ISH to Gfx (ISH2GFX_MSG21)—Offset 15E0h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.403 Outbound Inter Processor Messages 22 ISH to Gfx (ISH2GFX_MSG22)—Offset 15E4h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.404 Outbound Inter Processor Messages 23 ISH to Gfx (ISH2GFX_MSG23)—Offset 15E8h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.405 Outbound Inter Processor Messages 24 ISH to Gfx (ISH2GFX_MSG24)—Offset 15ECh

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.406 Outbound Inter Processor Messages 25 ISH to Gfx (ISH2GFX_MSG25)—Offset 15F0h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.407 Outbound Inter Processor Messages 26 ISH to Gfx (ISH2GFX_MSG26)—Offset 15F4h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.408 Outbound Inter Processor Messages 27 ISH to Gfx (ISH2GFX_MSG27)—Offset 15F8h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.409 Outbound Inter Processor Messages 28 ISH to Gfx (ISH2GFX_MSG28)—Offset 15FCh

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.410 Outbound Inter Processor Messages 29 ISH to Gfx (ISH2GFX_MSG29)—Offset 1600h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.411 Outbound Inter Processor Messages 30 ISH to Gfx (ISH2GFX_MSG30)—Offset 1604h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.412 Outbound Inter Processor Messages 31 ISH to Gfx (ISH2GFX_MSG31)—Offset 1608h

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.413 Outbound Inter Processor Messages 32 ISH to Gfx (ISH2GFX_MSG32)—Offset 160Ch

Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from ISH to GFX (MSG):** Inter-processor message registers for ISH core to communicate to the Gfx. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by Gfx.

20.414 Inbound Inter Processor Messages 1 Gfx to ISH (GFX2ISH_MSG1)—Offset 1610h

Inter-processor Message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from GFX to ISH (MSG):** Inter-processor Message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.415 Inbound Inter Processor Messages 2 Gfx to ISH (GFX2ISH_MSG2)—Offset 1614h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.416 Inbound Inter Processor Messages 3 Gfx to ISH (GFX2ISH_MSG3)—Offset 1618h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.417 Inbound Inter Processor Messages 4 Gfx to ISH (GFX2ISH_MSG4)—Offset 161Ch

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.418 Inbound Inter Processor Messages 5 Gfx to ISH (GFX2ISH_MSG5)—Offset 1620h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.419 Inbound Inter Processor Messages 6 Gfx to ISH (GFX2ISH_MSG6)—Offset 1624h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.420 Inbound Inter Processor Messages 7 Gfx to ISH (GFX2ISH_MSG7)—Offset 1628h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.421 Inbound Inter Processor Messages 8 Gfx to ISH (GFX2ISH_MSG8)—Offset 162Ch

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.422 Inbound Inter Processor Messages 9 Gfx to ISH (GFX2ISH_MSG9)—Offset 1630h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.423 Inbound Inter Processor Messages 10 Gfx to ISH (GFX2ISH_MSG10)—Offset 1634h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.424 Inbound Inter Processor Messages 11 Gfx to ISH (GFX2ISH_MSG11)—Offset 1638h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.425 Inbound Inter Processor Messages 12 Gfx to ISH (GFX2ISH_MSG12)—Offset 163Ch

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.426 Inbound Inter Processor Messages 13 Gfx to ISH (GFX2ISH_MSG13)—Offset 1640h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.427 Inbound Inter Processor Messages 14 Gfx to ISH (GFX2ISH_MSG14)—Offset 1644h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.428 Inbound Inter Processor Messages 15 Gfx to ISH (GFX2ISH_MSG15)—Offset 1648h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.429 Inbound Inter Processor Messages 16 Gfx to ISH (GFX2ISH_MSG16)—Offset 164Ch

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.430 Inbound Inter Processor Messages 17 Gfx to ISH (GFX2ISH_MSG17)—Offset 1650h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.431 Inbound Inter Processor Messages 18 Gfx to ISH (GFX2ISH_MSG18)—Offset 1654h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.432 Inbound Inter Processor Messages 19 Gfx to ISH (GFX2ISH_MSG19)—Offset 1658h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.433 Inbound Inter Processor Messages 20 Gfx to ISH (GFX2ISH_MSG20)—Offset 165Ch

Inter-processor Message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from GFX to ISH (MSG):** Inter-processor Message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.434 Inbound Inter Processor Messages 21 Gfx to ISH (GFX2ISH_MSG21)—Offset 1660h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.435 Inbound Inter Processor Messages 22 Gfx to ISH (GFX2ISH_MSG22)—Offset 1664h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.436 Inbound Inter Processor Messages 23 Gfx to ISH (GFX2ISH_MSG23)—Offset 1668h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.437 Inbound Inter Processor Messages 24 Gfx to ISH (GFX2ISH_MSG24)—Offset 166Ch

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.438 Inbound Inter Processor Messages 25 Gfx to ISH (GFX2ISH_MSG25)—Offset 1670h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.439 Inbound Inter Processor Messages 26 Gfx to ISH (GFX2ISH_MSG26)—Offset 1674h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.440 Inbound Inter Processor Messages 27 Gfx to ISH (GFX2ISH_MSG27)—Offset 1678h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.441 Inbound Inter Processor Messages 28 Gfx to ISH (GFX2ISH_MSG28)—Offset 167Ch

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.442 Inbound Inter Processor Messages 29 Gfx to ISH (GFX2ISH_MSG29)—Offset 1680h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.443 Inbound Inter Processor Messages 30 Gfx to ISH (GFX2ISH_MSG30)—Offset 1684h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.444 Inbound Inter Processor Messages 31 Gfx to ISH (GFX2ISH_MSG31)—Offset 1688h

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.445 Inbound Inter Processor Messages 32 Gfx to ISH (GFX2ISH_MSG32)—Offset 168Ch

Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from GFX to ISH (MSG):** Inter-processor message registers for Gfx to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Gfx to ISH. These registers are meant to be written by the Gfx and read by ISH.

20.446 Inbound Doorbell ISP to ISH (ISP2ISH_DOORBELL)—Offset 1690h

Inbound doorbell register, ISP core to interrupt ISH. Data 30:0 is 31 bit message payload used for backward compatibility. Software can use either this 31 bit message payload or 256 bit payload registers.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** ISP to ISH doorbell busy bit. When this bit is cleared, the ISH CPU is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** ISP to ISH doorbell 31 bit payload, 31 bit message payload for backward compatibility.

20.447 Outbound Doorbell ISH to ISP (ISH2ISP_DOORBELL)—Offset 1694h

Outbound doorbell register for the ISH to interrupt the ISP. Setting bit 31 of this register causes the ISP to receive a legacy INTA interrupt. Data 30:0 is 31 bit message payload used for backward compatibility.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** ISH to ISP doorbell busy bit. When this bit is cleared, the ISP is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** ISH to ISP doorbell 31 bit payload, 31 bit message payload for backward compatibility.

20.448 Inbound Doorbell Audio to ISH (AUDIO2ISH_DOORBELL)—Offset 1698h

Inbound doorbell register, audio core to interrupt ISH. Data 30:0 is 31 bit message payload used for backward compatibility. Software can use either this 31 bit message payload or 256 bit payload registers.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY)**
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** Audio to ISH doorbell 31 bit payload, 31 bit message payload for backward compatibility.

20.449 Outbound Doorbell ISH to Audio (ISH2AUDIO_DOORBELL)—Offset 169Ch

Outbound doorbell register for the ISH to interrupt the audio. Setting bit 31 of this register causes the audio to receive a legacy INTA interrupt. Data 30:0 is 31 bit message payload used for backward compatibility. Software can use either this 31 bit message payload or 256 bit payload registers.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** ISH to audio doorbell busy bit. When this bit is cleared, the audio is ready to accept a new message.
30:0	0h RW	**31 bit Payload (PAYLOAD_31BIT):** ISH to audio doorbell 31 bit Payload, 31 bit message payload for backward compatibility.

20.450 Inbound Doorbell GFX to ISH (GFX2ISH_DOORBELL)—Offset 16A0h

Inbound doorbell register, Gfx core to interrupt ISH. Data 30:0 is 31 bit message payload used for backward compatibility. Software can use either this 31 bit message payload or 256 bit payload registers.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** GFX to ISH doorbell busy bit. when this bit is cleared, the ISH CPU is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** GFX to ISH doorbell 31 bit payload, 31 bit message payload for backward compatibility.

20.451 Outbound Doorbell ISH to GFx (ISH2GFX_DOORBELL)—Offset 16A4h

Outbound doorbell register for the ISH to interrupt the Gfx. Setting bit 31 of this register causes the Gfx to receive a legacy INTA interrupt. Data 30:0 is 31 bit message payload used for backward compatibility. Software can use either this 31 bit message payload or 256 bit payload registers.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** ISH to GFX doorbell busy bit. When this bit is cleared, the GFX is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** ISH to GFX doorbell 31 bit payload, 31 bit message payload for backward compatibility.

20.452 CSME to ISH Control and Status (CSME2ISH_CTRL_STATUS)—Offset 16A8h

This is a 32 bit register which is set to 0 on a system reset or a wakeup from D3Cold. When CSME writes to any of these bits with 1, an interrupt is generated to mIA. The Interrupt is then cleared by mIA by writing 1`b1 to this register after the appropriate ISR is serviced.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/1S	**32 Bit Payload (CSME2ISH_CTRL_STATUS):** Control and status information transfer from CSME FW to ISH FW.

20.453 ISH IPC D0I3 Control (IPC_d0i3C_reg)—Offset 16D0h

This is a 32 bit register which is set to 0 on a system reset or a wakeup from D3Cold. When CSME writes to any of these bits with 1, an interrupt is generated to mIA. The Interrupt is then cleared by mIA by writing 1`b1 to this register after the appropriate ISR is serviced.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 8h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RO	**Reserved (RESERVED):** Reserved.
4	0h RO	**Interrupt Request Capable (ISH_IPC_d0i3C_reg_RSVD)**
3	1h RW/1C	**Restore Required (ISH_IPC_d0i3C_reg3):** D0i3 register for storing power related information RR: Restore Required bit.
2	0h RW	**D0i3 (ISH_IPC_d0i3C_reg2):** D0i3 register for storing power related information i3: D0i3 bit.
1	0h RW	**Interrupt Request (ISH_IPC_d0i3C_reg1):** D0i3 register for storing power related information IR: Interrupt Request bit.
0	0h RW/1C	**Command-In-Progress (ISH_IPC_d0i3C_reg0):** D0i3 register for storing power related information CIP: Command-In-Progress bit.

20.454 PMC to ISH Control and Status (ipc_pmc2ish_csr_reg)—Offset 16D4h

This is a 32 bit register which is set to 0 on a system reset or a wakeup from D3Cold. When PMC writes to any of these bits with 1, an interrupt is generated to mIA. The Interrupt is then cleared by mIA by writing 1`b1 to this register after the appropriate ISR is serviced.

Access Method

Type: MSG Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/1S	**32 Bit Payload (ISH_IPC_PMC2ISH_CSR_REG):** Control and status information transfer between ISH and PMC FWs and interrupt generation to mIA.

20.455 ISP to ISH Control and Status (ipc_isp2ish_csr_reg)—Offset 16D8h

This is a control and status register for enabling ISP to interrupt mIA incase of ISP writes to any of these bits of the register. The Interrupt is then cleared by mIA by writing 1`b1 to this register after the appropriate ISR is serviced.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/1S	**32 Bit Payload (ISH_IPC_ISP2ISH_CSR_REG):** Control and status information transfer between ISH and ISP FWs and interrupt generation to mIA.

20.456 Audio to ISH Control and Status (ipc_audio2ish_csr_reg)—Offset 16DCh

This is a control and status register for enabling audio to interrupt mIA incase of audio writes to any of these bits of the register. The Interrupt is then cleared by mIA by writing 1`b1 to this register after the appropriate ISR is serviced.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/1S	**32 Bit Payload (ISH_IPC_AUDIO2ISH_CSR_REG):** Control and status information transfer between ISH and audio FWs and interrupt generation to mIA.

20.457 Peripheral Interrupt Status (PISR)—Offset 0h

This register contains all the inbound and outbound interrupt status bits of the IPC registers. Interrupts ISH2HOST_Intr, ISH2SEC_Intr, ISH2PMC_Intr, HOST2ISH_intr, SEC2ISH_Intr and PMC2ISH_Intr are generated when PISR[x] & PIMR[x].

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:27	0h RO	**Reserved (RESERVED1):** Reserved.
26	0h RO	**Interrupt Status Bit for Audio CSR Register (PISR_AUDIO_CSR):** Audio to ISH CSR Ored interrupt: • 1: Interrupt is active. • 0: Interrupt is inactive.
25	0h RO	**Interrupt Status Bit for ISP CSR Register (PISR_ISP_CSR):** ISP to ISH CSR Ored interrupt: • 1: Interrupt is active. • 0: Interrupt is inactive.
24	0h RO	**Interrupt Status Bit for PMC CSR Register (PISR_PMC_CSR):** PMC to ISH CSR Ored interrupt: • 1: Interrupt is active. • 0: Interrupt is inactive.
23	0h RO	**Interrupt Status Bit for CSME CSR Register (PISR_CSME_CSR):** CSME to ISH CSR Ored interrupt: • 1: Interrupt is active. • 0: Interrupt is inactive.
22:12	0h RO	**Reserved (RESERVED0):** Reserved.
11	0h RO	**ISH to Audio Status (PISR_ISH2AUDIO):** Outbound IPC request from ISH to audio status: • 1: Interrupt is active. • 0: Interrupt is inactive.
10	0h RO	**ISH to ISP Status (PISR_ISH2ISP):** Outbound IPC request from ISH to ISP status: • 1: Interrupt is active. • 0: Interrupt is inactive.
9	0h RO	**ISH to GFX Status (PISR_ISH2GFX):** Outbound IPC request from ISH to GFX status: • 1: Interrupt is active. • 0: Interrupt is inactive.
8	0h RO	**Audio to ISH Status (PISR_AUDIO2ISH):** Inbound IPC request from audio to ISH status: • 1: Interrupt is active. • 0: Interrupt is inactive.
7	0h RO	**ISP to ISH Status (PISR_ISP2ISH):** Inbound IPC request from ISP to ISH status: • 1: Interrupt is active. • 0: Interrupt is inactive.
6	0h RO	**GFX to ISH Status (PISR_GFX2ISH):** Inbound IPC request from GFX to ISH status: • 1: Interrupt is active. • 0: Interrupt is inactive.
5	0h RO	**ISH to SEC Status (PISR_ISH2SEC):** Outbound IPC request from ISH to SEC status: • 1: Interrupt is active. • 0: Interrupt is inactive.
4	0h RO	**ISH to PMC Status (PISR_ISH2PMC):** Outbound IPC request from ISH to PMC status: • 1: Interrupt is active. • 0: Interrupt is inactive.
3	0h RO	**Reserved (RESERVED):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RO	**SEC to ISH Status (PISR_SEC2ISH):** Inbound IPC request from SEC to ISH status: • 1: Interrupt is active. • 0: Interrupt is inactive.
1	0h RO	**PMC to ISH Status (PISR_PMC2ISH):** Inbound IPC request from PMC to ISH status: • 1: Interrupt is active. • 0: Interrupt is inactive.
0	0h RO	**Host to ISH Status (PISR_HOST2ISH):** Inbound IPC request from host to ISH status: • 1: Interrupt is active. • 0: Interrupt is inactive.

20.458 Peripheral Interrupt Mask (PIMR)—Offset 4h

This register enables or disables inbound and outbound interrupts between ISH&HOST, ISH&PMC and ISH&PMC. The outbound interrupts cannot be masked by HOST, PMC & SEC since this register is not accessible by external agents like HOST, PMC & SEC.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RESERVED1):** Reserved.
27	0h RW	**H2IBCISC Interrupt Mask (H2IBCISC_IE):** Mask bit for H2IBCISC interrupt mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
26	0h RW	**Audio CSR Interrupt Mask (PIMR_AUDIO_CSR_MASK):** Mask bit for audio CSR interrupt mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
25	0h RW	**ISP CSR Interrupt Mask (PIMR_ISP_CSR_MASK):** Mask bit for ISP CSR interrupt mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
24	0h RW	**PMC CSR Interrupt Mask (PIMR_PMC_CSR_MASK):** Mask bit for PMC CSR interrupt mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
23	0h RW	**CSME CSR Interrupt Mask (PIMR_CSME_CSR_MASK):** Mask bit for CSME CSR interrupt mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
22	0h RW	**ISH to Audio Busy Clear Mask (PIMR_ISH2AUDIO_BUSY_CLEAR):** Mask bit for ISH to audio busy clear interrupt: • 1: Interrupt is unmasked. • 0: Interrupt is masked.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**ISH to ISP Busy Clear Mask (PIMR_ISH2ISP_BUSY_CLEAR):** Mask bit for ISH to ISP busy clear interrupt: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
20	0h RW	**ISH to GFX Busy Clear Mask (PIMR_ISH2GFX_BUSY_CLEAR):** Mask bit for ISH to GFX busy clear interrupt: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
19	0h RW	**ISH to Audio Mask (PIMR_ISH2AUDIO):** Outbound IPC request from ISH to audio mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
18	0h RW	**ISH to ISP Mask (PIMR_ISH2ISP):** Outbound IPC request from ISH to ISP mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
17	0h RW	**ISH to GFX Mask (PIMR_ISH2GFX):** Outbound IPC request from ISH to GFX mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
16	0h RW	**Audio to ISH Mask (PIMR_AUDIO2ISH):** Inbound IPC request from audio to ISH mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
15	0h RW	**ISP to ISH Mask (PIMR_ISP2ISH):** Inbound IPC request from ISP to ISH mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
14	0h RW	**GFX to ISH Mask (PIMR_GFX2ISH):** Inbound IPC request from GFX to ISH mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
13	0h RW	**ISH to SEC Busy Clear Mask (PIMR_ISH2SEC_BUSY_CLEAR):** Mask bit for ISH to SEC busy clear interrupt: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
12	0h RW	**ISH to PMC Busy Clear Mask (PIMR_ISH2PMC_BUSY_CLEAR):** Mask bit for ISH to PMC busy clear interrupt: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
11	0h RW	**ISH to Host Busy Clear Mask (PIMR_ISH2HOST_BUSY_CLEAR):** Mask bit for ISH to host busy clear interrupt: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
10:6	0h RO	**Reserved (RESERVED0):** Reserved.
5	0h RW	**ISH to SEC Mask (PIMR_ISH2SEC):** Outbound IPC request from ISH to SEC mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
4	0h RW	**ISH to PMC Mask (PIMR_ISH2PMC):** Outbound IPC request from ISH to PMC mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
3	0h RO	**Reserved (RESERVED):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**SEC to ISH Mask (PIMR_SEC2ISH):** Inbound IPC request from SEC to ISH mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
1	0h RW	**PMC to ISH Mask (PIMR_PMC2ISH):** Inbound IPC request from PMC to ISH mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
0	0h RW	**Host to ISH Mask (PIMR_HOST2ISH):** Inbound IPC request from host to ISH mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.

20.459 Host Peripheral Interrupt Mask (HOST_PIMR)—Offset 8h

This register is for masking the interrupt, caused by setting the DB bit, for IPC from mIA to host.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Reserved (RESERVED1):** Reserved.
8	0h RW	**Host to ISH Busy Clear Mask (PIMR_HOST2ISH_BUSY_CLEAR):** Mask bit for host to ISH busy clear interrupt: • 1: Interrupt is unmasked. • 0: Interrupt is masked.
7:1	0h RO	**Reserved (RESERVED0):** Reserved.
0	0h RW	**Mask Interrupt Caused By ISH to Host Doorbell (PIMR_ISH2HOST_IPC_REG):** Outbound IPC request from ISH to host mask: • 1: Interrupt is unmasked. • 0: Interrupt is masked.

20.460 HOST Peripheral Interrupt Status (HOST_PISR)—Offset Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Reserved (RESERVED1):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW/1C	**Host to ISH Busy Clear Interrupt Status Bit (PISR_HOST2ISH_BUSY_CLEAR):** 1: Interrupt is active.
7:1	0h RO	**Reserved (RESERVED0):** Reserved.
0	0h RO	**ISH to HOST DB Interrupt Status Bit (PISR_ISH2HOST_IPC_REG)**

20.461 Reserved (RESERVED2)—Offset 10h

Reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**Reserved (RESERVED2):** Reserved.

20.462 Reserved (RESERVED3)—Offset 14h

Reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**Reserved (RESERVED3):** Reserved.

20.463 Reserved (RESERVED4)—Offset 18h

Reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**Reserved (RESERVED4):** Reserved.

20.464 Reserved (RESERVED5)—Offset 1Ch

Reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**Reserved (RESERVED5):** Reserved.

20.465 Reserved (RESERVED6)—Offset 20h

Reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**Reserved (RESERVED6):** Reserved.

20.466 Reserved (RESERVED7)—Offset 24h

Reserved.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**Reserved (RESERVED7):** Reserved.

20.467 Reserved (RESERVED8)—Offset 28h

Reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**Reserved (RESERVED8):** Reserved.

20.468 Reserved (RESERVED9)—Offset 2Ch

Reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**Reserved (RESERVED9):** Reserved.

20.469 Reserved (RESERVED10)—Offset 30h

Reserved.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**Reserved (RESERVED10):** Reserved.

20.470 ISH Host Firmware Status (ISH_HOST_FWSTS)—Offset 34h

This is a 32 bit register which is set to 0x0000 on a system reset or a wakeup from D3Cold. As the firmware comes up after the reset or power cycle it sets bits of this register to 1`b1 to indicate its status.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**ISH Host Firmware Status (ISH_HOST_FWSTS)**

20.471 Host Communication (HOST_COMM)—Offset 38h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Host Communication Register (HOST_COMM)**

20.472 ISH SEC Firmware Status (ISH_SEC_FWSTS)—Offset 3Ch

This is a 32 bit register which is set to 0x0000 on a system reset or a wakeup from D3Cold. As the firmware comes up after the reset or power cycle it sets bits of this register to 1`b1 to indicate its status.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**ISH SEC Firmware Status (ISH_SEC_FWSTS)**

20.473 SEC Communication (SEC_COMM)—Offset 40h

This is a 32 bit register which is set to 0x0000 on a system reset or a wakeup from D3Cold. The SEC sets bits of this register to 1`b1 to communicate with the ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SEC Communication (SEC_COMM):** This is a 32 bit register which is set to 0x0000 on a system reset or a wakeup from D3Cold. The SEC sets bits of this register to 1`b1 to communicate with the ISH.

20.474 ISH Reset (ISH_RST)—Offset 44h

This register is reserved for SEC. SEC should not write to this register. The reset register is a 32 bit register that can be written by ISH Minute IA. The Host cannot read or write this bit. Setting bit 1 will cause MinuteIA to reset.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RO	**Reserved (RESERVED):** Reserved.
0	0h RW	**Reset Bit (RESET_BIT):** ISH Reset Register: • 1: Reset active. • 0: Reset inactive.

20.475 Inbound Doorbell HOST to ISH (HOST2ISH_DOORBELL)—Offset 48h

Inbound doorbell register, host core to interrupt ISH. Data 30:0 is 31 bit message payload used for backward compatibility.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** When this bit is cleared, the ISH CPU is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** 31 bit message payload for backward compatibility.

20.476 Inbound Doorbell PMC to ISH (PMC2ISH_DOORBELL)—Offset 4Ch

Inbound doorbell register, PMC to interrupt ISH Data 30:0 is 31 bit message payload used for backward compatibility.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** When this bit is cleared, the ISH CPU is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** 31 bit message payload for backward compatibility.

20.477 Inbound Doorbell SEC to ISH (SEC2ISH_DOORBELL)—Offset 50h

Inbound doorbell register, SEC to interrupt ISH Data 30:0 is 31 bit message payload used for backward compatibility.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** When this bit is cleared, the ISH CPU is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** 31 bit message payload for backward compatibility.

20.478 Outbound Doorbell ISH to Host (ISH2HOST_DOORBELL)—Offset 54h

Outbound doorbell register for the ISH to interrupt the host. Setting bit 31 of this register causes the host to receive a IRQn interrupt. Data 30:0 is 31 bit message payload used for backward compatibility.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** When this bit is cleared, the host CPU is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** 31 bit message payload for backward compatibility.

20.479 Outbound Doorbell ISH to PMC (ISH2PMC_DOORBELL)—Offset 58h

Outbound doorbell register for the ISH to interrupt the PMC. Setting bit 31 of this register causes the PMC to receive interrupt on OOB. Data 30:0 is 31 bit message payload used for backward compatibility.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** When this bit is cleared, the PMC CPU is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** 31 bit message payload for backward compatibility.

20.480 Outbound Doorbell ISH to SEC (ISH2SEC_DOORBELL)—Offset 5Ch

Outbound doorbell register for the ISH to interrupt the SEC. Setting bit 31 of this register causes the SEC to receive interrupt on OOB Data 30:0 is 31 bit message payload used for backward compatibility.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** When this bit is cleared, the SEC CPU is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** 31 bit message payload for backward compatibility.

20.481 Outbound Inter-processor Messages 1 ISH to Host (ISH2HOST_MSG1)—Offset 60h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.482 Outbound Inter-processor Messages 2 ISH to Host (ISH2HOST_MSG2)—Offset 64h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to Host.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to Host.

20.483 Outbound Inter-processor Messages 3 ISH to Host (ISH2HOST_MSG3)—Offset 68h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.484 Outbound Inter-processor Messages 4 ISH to Host (ISH2HOST_MSG4)—Offset 6Ch

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.485 Outbound Inter-processor Messages 5 ISH to Host (ISH2HOST_MSG5)—Offset 70h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.486 Outbound Inter-processor Messages 6 ISH to Host (ISH2HOST_MSG6)—Offset 74h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.487 Outbound Inter-processor Messages 7 ISH to Host (ISH2HOST_MSG7)—Offset 78h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.488 Outbound Inter-processor Messages 8 ISH to Host (ISH2HOST_MSG8)—Offset 7Ch

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.489 Outbound Inter-processor Messages 9 ISH to Host (ISH2HOST_MSG9)—Offset 80h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.490 Outbound Inter-processor Messages 10 ISH to Host (ISH2HOST_MSG10)—Offset 84h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.491 Outbound Inter-processor Messages 11 ISH to Host (ISH2HOST_MSG11)—Offset 88h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.492 Outbound Inter-processor Messages 12 ISH to Host (ISH2HOST_MSG12)—Offset 8Ch

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.493 Outbound Inter-processor Messages 13 ISH to Host (ISH2HOST_MSG13)—Offset 90h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.494 Outbound Inter-processor Messages 14 ISH to Host (ISH2HOST_MSG14)—Offset 94h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.495 Outbound Inter-processor Messages 15 ISH to Host (ISH2HOST_MSG15)—Offset 98h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.496 Outbound Inter-processor Messages 16 ISH to Host (ISH2HOST_MSG16)—Offset 9Ch

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.497 Outbound Inter-processor Messages 17 ISH to Host (ISH2HOST_MSG17)—Offset A0h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.498 Outbound Inter-processor Messages 18 ISH to Host (ISH2HOST_MSG18)—Offset A4h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.499 Outbound Inter-processor Messages 19 ISH to Host (ISH2HOST_MSG19)—Offset A8h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.500 Outbound Inter-processor Messages 20 ISH to Host (ISH2HOST_MSG20)—Offset ACh

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.501 Outbound Inter-processor Messages 21 ISH to Host (ISH2HOST_MSG21)—Offset B0h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.502 Outbound Inter-processor Messages 22 ISH to Host (ISH2HOST_MSG22)—Offset B4h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.503 Outbound Inter-processor Messages 23 ISH to Host (ISH2HOST_MSG23)—Offset B8h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.504 Outbound Inter-processor Messages 24 ISH to Host (ISH2HOST_MSG24)—Offset BCh

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.505 Outbound Inter-processor Messages 25 ISH to Host (ISH2HOST_MSG25)—Offset C0h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.506 Outbound Inter-processor Messages 26 ISH to Host (ISH2HOST_MSG26)—Offset C4h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.507 Outbound Inter-processor Messages 27 ISH to Host (ISH2HOST_MSG27)—Offset C8h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.508 Outbound Inter-processor Messages 28 ISH to Host (ISH2HOST_MSG28)—Offset CCh

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.509 Outbound Inter-processor Messages 29 ISH to Host (ISH2HOST_MSG29)—Offset D0h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.510 Outbound Inter-processor Messages 30 ISH to Host (ISH2HOST_MSG30)—Offset D4h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.511 Outbound Inter-processor Messages 31 ISH to Host (ISH2HOST_MSG31)—Offset D8h

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.512 Outbound Inter-processor Messages 32 ISH to Host (ISH2HOST_MSG32)—Offset DCh

Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from ISH to Host (MSG):** Inter-process message registers for ISH core to communicate to the host. These are thirty-two 32 bit registers that hold the message payload from the ISH to host.

20.513 Inbound Inter-processor Messages 1 Host to ISH (HOST2ISH_MSG1)—Offset E0h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.514 Inbound Inter-processor Messages 2 Host to ISH (HOST2ISH_MSG2)—Offset E4h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.515 Inbound Inter-processor Messages 3 Host to ISH (HOST2ISH_MSG3)—Offset E8h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.516 Inbound Inter-processor Messages 4 Host to ISH (HOST2ISH_MSG4)—Offset ECh

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.517 Inbound Inter-processor Messages 5 Host to ISH (HOST2ISH_MSG5)—Offset F0h

Inter-process Message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from Host to ISH (MSG):** Inter-process Message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the Host to ISH.

20.518 Inbound Inter-processor Messages 6 Host to ISH (HOST2ISH_MSG6)—Offset F4h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.519 Inbound Inter-processor Messages 7 Host to ISH (HOST2ISH_MSG7)—Offset F8h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.520 Inbound Inter-processor Messages 8 Host to ISH (HOST2ISH_MSG8)—Offset FCh

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.521 Inbound Inter-processor Messages 9 Host to ISH (HOST2ISH_MSG9)—Offset 100h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.522 Inbound Inter-processor Messages 10 Host to ISH (HOST2ISH_MSG10)—Offset 104h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.523 Inbound Inter-processor Messages 11 Host to ISH (HOST2ISH_MSG11)—Offset 108h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.524 Inbound Inter-processor Messages 12 Host to ISH (HOST2ISH_MSG12)—Offset 10Ch

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.525 Inbound Inter-processor Messages 13 Host to ISH (HOST2ISH_MSG13)—Offset 110h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.526 Inbound Inter-processor Messages 14 Host to ISH (HOST2ISH_MSG14)—Offset 114h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.527 Inbound Inter-processor Messages 15 Host to ISH (HOST2ISH_MSG15)—Offset 118h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.528 Inbound Inter-processor Messages 16 Host to ISH (HOST2ISH_MSG16)—Offset 11Ch

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.529 Inbound Inter-processor Messages 17 Host to ISH (HOST2ISH_MSG17)—Offset 120h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.530 Inbound Inter-processor Messages 18 Host to ISH (HOST2ISH_MSG18)—Offset 124h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.531 Inbound Inter-processor Messages 19 Host to ISH (HOST2ISH_MSG19)—Offset 128h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.532 Inbound Inter-processor Messages 20 Host to ISH (HOST2ISH_MSG20)—Offset 12Ch

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.533 Inbound Inter-processor Messages 21 Host to ISH (HOST2ISH_MSG21)—Offset 130h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.534 Inbound Inter-processor Messages 22 Host to ISH (HOST2ISH_MSG22)—Offset 134h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.535 Inbound Inter-processor Messages 23 Host to ISH (HOST2ISH_MSG23)—Offset 138h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.536 Inbound Inter-processor Messages 24 Host to ISH (HOST2ISH_MSG24)—Offset 13Ch

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.537 Inbound Inter-processor Messages 25 Host to ISH (HOST2ISH_MSG25)—Offset 140h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.538 Inbound Inter-processor Messages 26 Host to ISH (HOST2ISH_MSG26)—Offset 144h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.539 Inbound Inter-processor Messages 27 Host to ISH (HOST2ISH_MSG27)—Offset 148h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.540 Inbound Inter-processor Messages 28 Host to ISH (HOST2ISH_MSG28)—Offset 14Ch

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.541 Inbound Inter-processor Messages 29 Host to ISH (HOST2ISH_MSG29)—Offset 150h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.542 Inbound Inter-processor Messages 30 Host to ISH (HOST2ISH_MSG30)—Offset 154h

Inter-process message registers for Host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from Host to ISH (MSG):** Inter-process message registers for Host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.543 Inbound Inter-processor Messages 31 Host to ISH (HOST2ISH_MSG31)—Offset 158h

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.544 Inbound Inter-processor Messages 32 Host to ISH (HOST2ISH_MSG32)—Offset 15Ch

Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from Host to ISH (MSG):** Inter-process message registers for host to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the host to ISH.

20.545 Outbound Inter-processor Messages 1 ISH to SEC (ISH2SEC_MSG1)—Offset 160h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.546 Outbound Inter-processor Messages 2 ISH to SEC (ISH2SEC_MSG2)—Offset 164h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.547 Outbound Inter-processor Messages 3 ISH to SEC (ISH2SEC_MSG3)—Offset 168h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.548 Outbound Inter-processor Messages 4 ISH to SEC (ISH2SEC_MSG4)—Offset 16Ch

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.549 Outbound Inter-processor Messages 5 ISH to SEC (ISH2SEC_MSG5)—Offset 170h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.550 Outbound Inter-processor Messages 6 ISH to SEC (ISH2SEC_MSG6)—Offset 174h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.551 Outbound Inter-processor Messages 7 ISH to SEC (ISH2SEC_MSG7)—Offset 178h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.552 Outbound Inter-processor Messages 8 ISH to SEC (ISH2SEC_MSG8)—Offset 17Ch

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.553 Outbound Inter-processor Messages 9 ISH to SEC (ISH2SEC_MSG9)—Offset 180h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.554 Outbound Inter-processor Messages 10 ISH to SEC (ISH2SEC_MSG10)—Offset 184h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.555 Outbound Inter-processor Messages 11 ISH to SEC (ISH2SEC_MSG11)—Offset 188h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.556 Outbound Inter-processor Messages 12 ISH to SEC (ISH2SEC_MSG12)—Offset 18Ch

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.557 Outbound Inter-processor Messages 13 ISH to SEC (ISH2SEC_MSG13)—Offset 190h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.558 Outbound Inter-processor Messages 14 ISH to SEC (ISH2SEC_MSG14)—Offset 194h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.559 Outbound Inter-processor Messages 15 ISH to SEC (ISH2SEC_MSG15)—Offset 198h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.560 Outbound Inter-processor Messages 16 ISH to SEC (ISH2SEC_MSG16)—Offset 19Ch

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.561 Outbound Inter-processor Messages 17 ISH to SEC (ISH2SEC_MSG17)—Offset 1A0h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.562 Outbound Inter-processor Messages 18 ISH to SEC (ISH2SEC_MSG18)—Offset 1A4h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.563 Outbound Inter-processor Messages 19 ISH to SEC (ISH2SEC_MSG19)—Offset 1A8h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.564 Outbound Inter-processor Messages 20 ISH to SEC (ISH2SEC_MSG20)—Offset 1ACh

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.565 Outbound Inter-processor Messages 21 ISH to SEC (ISH2SEC_MSG21)—Offset 1B0h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.566 Outbound Inter-processor Messages 22 ISH to SEC (ISH2SEC_MSG22)—Offset 1B4h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.567 Outbound Inter-processor Messages 23 ISH to SEC (ISH2SEC_MSG23)—Offset 1B8h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.568 Outbound Inter-processor Messages 24 ISH to SEC (ISH2SEC_MSG24)—Offset 1BCh

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.569 Outbound Inter-processor Messages 25 ISH to SEC (ISH2SEC_MSG25)—Offset 1C0h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.570 Outbound Inter-processor Messages 26 ISH to SEC (ISH2SEC_MSG26)—Offset 1C4h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.571 Outbound Inter-processor Messages 27 ISH to SEC (ISH2SEC_MSG27)—Offset 1C8h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.572 Outbound Inter-processor Messages 28 ISH to SEC (ISH2SEC_MSG28)—Offset 1CCh

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.573 Outbound Inter-processor Messages 29 ISH to SEC (ISH2SEC_MSG29)—Offset 1D0h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.574 Outbound Inter-processor Messages 30 ISH to SEC (ISH2SEC_MSG30)—Offset 1D4h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.575 Outbound Inter-processor Messages 31 ISH to SEC (ISH2SEC_MSG31)—Offset 1D8h

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.576 Outbound Inter-processor Messages 32 ISH to SEC (ISH2SEC_MSG32)—Offset 1DCh

Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from ISH to SEC (MSG):** Inter-process message registers for ISH core to communicate to the SEC. These are thirty-two 32 bit registers that hold the message payload from the ISH to SEC.

20.577 Inbound Inter-processor Messages 1 SEC to ISH (SEC2ISH_MSG1)—Offset 1E0h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.578 Inbound Inter-processor Messages 2 SEC to ISH (SEC2ISH_MSG2)—Offset 1E4h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.579 Inbound Inter-processor Messages 3 SEC to ISH (SEC2ISH_MSG3)—Offset 1E8h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.580 Inbound Inter-processor Messages 4 SEC to ISH (SEC2ISH_MSG4)—Offset 1ECh

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.581 Inbound Inter-processor Messages 5 SEC to ISH (SEC2ISH_MSG5)—Offset 1F0h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.582 Inbound Inter-processor Messages 6 SEC to ISH (SEC2ISH_MSG6)—Offset 1F4h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.583 Inbound Inter-processor Messages 7 SEC to ISH (SEC2ISH_MSG7)—Offset 1F8h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.584 Inbound Inter-processor Messages 8 SEC to ISH (SEC2ISH_MSG8)—Offset 1FCh

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.585 Inbound Inter-processor Messages 9 SEC to ISH (SEC2ISH_MSG9)—Offset 200h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.586 Inbound Inter-processor Messages 10 SEC to ISH (SEC2ISH_MSG10)—Offset 204h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.587 Inbound Inter-processor Messages 11 SEC to ISH (SEC2ISH_MSG11)—Offset 208h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.588 Inbound Inter-processor Messages 12 SEC to ISH (SEC2ISH_MSG12)—Offset 20Ch

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.589 Inbound Inter-processor Messages 13 SEC to ISH (SEC2ISH_MSG13)—Offset 210h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message13 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.590 Inbound Inter-processor Messages 14 SEC to ISH (SEC2ISH_MSG14)—Offset 214h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.591 Inbound Inter-processor Messages 15 SEC to ISH (SEC2ISH_MSG15)—Offset 218h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.592 Inbound Inter-processor Messages 16 SEC to ISH (SEC2ISH_MSG16)—Offset 21Ch

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.593 Inbound Inter-processor Messages 17 SEC to ISH (SEC2ISH_MSG17)—Offset 220h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.594 Inbound Inter-processor Messages 18 SEC to ISH (SEC2ISH_MSG18)—Offset 224h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.595 Inbound Inter-processor Messages 19 SEC to ISH (SEC2ISH_MSG19)—Offset 228h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.596 Inbound Inter-processor Messages 20 SEC to ISH (SEC2ISH_MSG20)—Offset 22Ch

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.597 Inbound Inter-processor Messages 21 SEC to ISH (SEC2ISH_MSG21)—Offset 230h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.598 Inbound Inter-processor Messages 22 SEC to ISH (SEC2ISH_MSG22)—Offset 234h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.599 Inbound Inter-processor Messages 23 SEC to ISH (SEC2ISH_MSG23)—Offset 238h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.600 Inbound Inter-processor Messages 24 SEC to ISH (SEC2ISH_MSG24)—Offset 23Ch

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.601 Inbound Inter-processor Messages 25 SEC to ISH (SEC2ISH_MSG25)—Offset 240h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.602 Inbound Inter-processor Messages 26 SEC to ISH (SEC2ISH_MSG26)—Offset 244h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.603 Inbound Inter-processor Messages 27 SEC to ISH (SEC2ISH_MSG27)—Offset 248h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.604 Inbound Inter-processor Messages 28 SEC to ISH (SEC2ISH_MSG28)—Offset 24Ch

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.605 Inbound Inter-processor Messages 29 SEC to ISH (SEC2ISH_MSG29)—Offset 250h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.606 Inbound Inter-processor Messages 30 SEC to ISH (SEC2ISH_MSG30)—Offset 254h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.607 Inbound Inter-processor Messages 31 SEC to ISH (SEC2ISH_MSG31)—Offset 258h

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.608 Inbound Inter-processor Messages 32 SEC to ISH (SEC2ISH_MSG32)—Offset 25Ch

Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from SEC to ISH (MSG):** Inter-process message registers for SEC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the SEC to ISH.

20.609 Outbound Inter-processor Messages 1 ISH to PMC (ISH2PMC_MSG1)—Offset 260h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.610 Outbound Inter-processor Messages 2 ISH to PMC (ISH2PMC_MSG2)—Offset 264h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.611 Outbound Inter-processor Messages 3 ISH to PMC (ISH2PMC_MSG3)—Offset 268h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.612 Outbound Inter-processor Messages 4 ISH to PMC (ISH2PMC_MSG4)—Offset 26Ch

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.613 Outbound Inter-processor Messages 5 ISH to PMC (ISH2PMC_MSG5)—Offset 270h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.614 Outbound Inter-processor Messages 6 ISH to PMC (ISH2PMC_MSG6)—Offset 274h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.615 Outbound Inter-processor Messages 7 ISH to PMC (ISH2PMC_MSG7)—Offset 278h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.616 Outbound Inter-processor Messages 8 ISH to PMC (ISH2PMC_MSG8)—Offset 27Ch

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.617 Outbound Inter-processor Messages 9 ISH to PMC (ISH2PMC_MSG9)—Offset 280h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.618 Outbound Inter-processor Messages 10 ISH to PMC (ISH2PMC_MSG10)—Offset 284h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.619 Outbound Inter-processor Messages 11 ISH to PMC (ISH2PMC_MSG11)—Offset 288h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.620 Outbound Inter-processor Messages 12 ISH to PMC (ISH2PMC_MSG12)—Offset 28Ch

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.621 Outbound Inter-processor Messages 13 ISH to PMC (ISH2PMC_MSG13)—Offset 290h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.622 Outbound Inter-processor Messages 14 ISH to PMC (ISH2PMC_MSG14)—Offset 294h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.623 Outbound Inter-processor Messages 15 ISH to PMC (ISH2PMC_MSG15)—Offset 298h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.624 Outbound Inter-processor Messages 16 ISH to PMC (ISH2PMC_MSG16)—Offset 29Ch

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.625 Outbound Inter-processor Messages 17 ISH to PMC (ISH2PMC_MSG17)—Offset 2A0h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.626 Outbound Inter-processor Messages 18 ISH to PMC (ISH2PMC_MSG18)—Offset 2A4h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.627 Outbound Inter-processor Messages 19 ISH to PMC (ISH2PMC_MSG19)—Offset 2A8h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.628 Outbound Inter-processor Messages 20 ISH to PMC (ISH2PMC_MSG20)—Offset 2ACh

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.629 Outbound Inter-processor Messages 21 ISH to PMC (ISH2PMC_MSG21)—Offset 2B0h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.630 Outbound Inter-processor Messages 22 ISH to PMC (ISH2PMC_MSG22)—Offset 2B4h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.631 Outbound Inter-processor Messages 23 ISH to PMC (ISH2PMC_MSG23)—Offset 2B8h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.632 Outbound Inter-processor Messages 24 ISH to PMC (ISH2PMC_MSG24)—Offset 2BCh

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.633 Outbound Inter-processor Messages 25 ISH to PMC (ISH2PMC_MSG25)—Offset 2C0h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.634 Outbound Inter-processor Messages 26 ISH to PMC (ISH2PMC_MSG26)—Offset 2C4h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.635 Outbound Inter-processor Messages 27 ISH to PMC (ISH2PMC_MSG27)—Offset 2C8h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.636 Outbound Inter-processor Messages 28 ISH to PMC (ISH2PMC_MSG28)—Offset 2CCh

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.637 Outbound Inter-processor Messages 29 ISH to PMC (ISH2PMC_MSG29)—Offset 2D0h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.638 Outbound Inter-processor Messages 30 ISH to PMC (ISH2PMC_MSG30)—Offset 2D4h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.639 Outbound Inter-processor Messages 31 ISH to PMC (ISH2PMC_MSG31)—Offset 2D8h

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.640 Outbound Inter-processor Messages 32 ISH to PMC (ISH2PMC_MSG32)—Offset 2DCh

Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from ISH to PMC (MSG):** Inter-process message registers for ISH core to communicate to the PMC. These are thirty-two 32 bit registers that hold the message payload from the ISH to PMC.

20.641 Inbound Inter-processor Messages 1 PMC to ISH (PMC2ISH_MSG1)—Offset 2E0h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.642 Inbound Inter-processor Messages 2 PMC to ISH (PMC2ISH_MSG2)—Offset 2E4h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.643 Inbound Inter-processor Messages 3 PMC to ISH (PMC2ISH_MSG3)—Offset 2E8h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.644 Inbound Inter-processor Messages 4 PMC to ISH (PMC2ISH_MSG4)—Offset 2ECh

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.645 Inbound Inter-processor Messages 5 PMC to ISH (PMC2ISH_MSG5)—Offset 2F0h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.646 Inbound Inter-processor Messages 6 PMC to ISH (PMC2ISH_MSG6)—Offset 2F4h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.647 Inbound Inter-processor Messages 7 PMC to ISH (PMC2ISH_MSG7)—Offset 2F8h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.648 Inbound Inter-processor Messages 8 PMC to ISH (PMC2ISH_MSG8)—Offset 2FCh

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.649 Inbound Inter-processor Messages 9 PMC to ISH (PMC2ISH_MSG9)—Offset 300h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.650 Inbound Inter-processor Messages 10 PMC to ISH (PMC2ISH_MSG10)—Offset 304h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.651 Inbound Inter-processor Messages 11 PMC to ISH (PMC2ISH_MSG11)—Offset 308h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.652 Inbound Inter-processor Messages 12 PMC to ISH (PMC2ISH_MSG12)—Offset 30Ch

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.653 Inbound Inter-processor Messages 13 PMC to ISH (PMC2ISH_MSG13)—Offset 310h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.654 Inbound Inter-processor Messages 14 PMC to ISH (PMC2ISH_MSG14)—Offset 314h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.655 Inbound Inter-processor Messages 15 PMC to ISH (PMC2ISH_MSG15)—Offset 318h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.656 Inbound Inter-processor Messages 16 PMC to ISH (PMC2ISH_MSG16)—Offset 31Ch

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.657 Inbound Inter-processor Messages 17 PMC to ISH (PMC2ISH_MSG17)—Offset 320h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.658 Inbound Inter-processor Messages 18 PMC to ISH (PMC2ISH_MSG18)—Offset 324h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.659 Inbound Inter-processor Messages 19 PMC to ISH (PMC2ISH_MSG19)—Offset 328h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.660 Inbound Inter-processor Messages 20 PMC to ISH (PMC2ISH_MSG20)—Offset 32Ch

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.661 Inbound Inter-processor Messages 21 PMC to ISH (PMC2ISH_MSG21)—Offset 330h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.662 Inbound Inter-processor Messages 22 PMC to ISH (PMC2ISH_MSG22)—Offset 334h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.663 Inbound Inter-processor Messages 23 PMC to ISH (PMC2ISH_MSG23)—Offset 338h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.664 Inbound Inter-processor Messages 24 PMC to ISH (PMC2ISH_MSG24)—Offset 33Ch

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.665 Inbound Inter-processor Messages 25 PMC to ISH (PMC2ISH_MSG25)—Offset 340h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.666 Inbound Inter-processor Messages 26 PMC to ISH (PMC2ISH_MSG26)—Offset 344h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.667 Inbound Inter-processor Messages 27 PMC to ISH (PMC2ISH_MSG27)—Offset 348h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.668 Inbound Inter-processor Messages 28 PMC to ISH (PMC2ISH_MSG28)—Offset 34Ch

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.669 Inbound Inter-processor Messages 29 PMC to ISH (PMC2ISH_MSG29)—Offset 350h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from PMC to ISH. (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.670 Inbound Inter-processor Messages 30 PMC to ISH (PMC2ISH_MSG30)—Offset 354h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.671 Inbound Inter-processor Messages 31 PMC to ISH (PMC2ISH_MSG31)—Offset 358h

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.672 Inbound Inter-processor Messages 32 PMC to ISH (PMC2ISH_MSG32)—Offset 35Ch

Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from PMC to ISH (MSG):** Inter-process message registers for PMC to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the PMC to ISH.

20.673 Remap 0 (REMAP0)—Offset 360h

At boot time, the host can write these registers with any dynamically allocated address spaces that ISH will have to access.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Remap 0 (REMAP0)**

20.674 Remap 1 (REMAP1)—Offset 364h

At boot time, the host can write these registers with any dynamically allocated address spaces that ISH will have to access.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Remap 1 (REMAP1)**

20.675 Remap 2 (REMAP2)—Offset 368h

At boot time, the host can write these registers with any dynamically allocated address spaces that ISH will have to access.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Remap 2 (REMAP2)**

20.676 Remap 3 (REMAP3)—Offset 36Ch

At boot time, the host can write these registers with any dynamically allocated address spaces that ISH will have to access.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Remap 3 (REMAP3)**

20.677 Remap 4 (REMAP4)—Offset 370h

At boot time, the host can write these registers with any dynamically allocated address spaces that ISH will have to access.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Remap 4 (REMAP4)**

20.678 Remap 5 (REMAP5)—Offset 374h

At boot time, the host can write these registers with any dynamically allocated address spaces that ISH will have to access.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Remap 5 (REMAP5)**

20.679 ISH IPC Busy Clear (ISH_IPC_BUSY_CLEAR)—Offset 378h

This register holds the status of the ISH IPC busy clear interrupts. ISH IPC busy clear interrupt is set when busy bit of respective outbound doorbell register gets cleared and interrupt is cleared when ISH writes CAN1EM to respective bit in ISH IPC busy clear register.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RESERVED0):** Reserved.
16	0h RW/1C	**ISH to SEC Busy Clear Status (ISH2SEC_BUSY_CLEAR):** Busy clear interrupt bit of ISH to SEC IPC: • 1: Interrupt active. • 0: Interrupt inactive.
15:9	0h RO	**Reserved (RESERVED1):** Reserved.
8	0h RW/1C	**ISH to PMC Busy Clear Status (ISH2PMC_BUSY_CLEAR):** Busy clear interrupt bit of ISH to PMC IPC: • 1: Interrupt active. • 0: Interrupt inactive.
7:4	0h RO	**Reserved (RESERVED2):** Reserved.
3	0h RW/1C	**ISH to Audio Busy Clear Status (ISH2AUDIO_BUSY_CLEAR):** Busy clear interrupt bit of ISH to audio IPC: • 1: Interrupt active. • 0: Interrupt inactive.
2	0h RW/1C	**ISH to ISP Busy Clear Status (ISH2ISP_BUSY_CLEAR):** Busy clear interrupt bit of ISH to ISP IPC: • 1: Interrupt active. • 0: Interrupt inactive.
1	0h RW/1C	**ISH to GFX Busy Clear Status (ISH2GFX_BUSY_CLEAR):** Busy clear interrupt bit of ISH to GFX IPC: • 1: Interrupt active. • 0: Interrupt inactive.
0	0h RW/1C	**ISH to Host Busy Clear Status (ISH2HOST_BUSY_CLEAR):** Busy clear interrupt bit of ISH to host IPC: • 1: Interrupt active. • 0: Interrupt inactive.

20.680 ISH Fuse (ISH_FUSE)—Offset 37Ch

This register mirrors the ISH fuse bits.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RESERVED):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
3:0	0h RO	**ISH Fuse (ISH_FUSE)**

20.681 UMA Base Address LSB (UMA_RANGE_LOWER_0)—Offset 380h

This register should be accessed only in ISH3.0. This is an address range register, holds 32 bit LSB value of the address for UMA base range.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: FFFFF001h

Bit Range	Default & Access	Field Name (ID): Description
31:12	FFFFFh RW	**UMA Memory Base Address LSB (UMA_MEMORY_BASE_ADDR_LSB)**
11:1	0h RO	**NUL Memory Size (SIZE):** Hard wired to 0 indicate 4 KB window granularity.
0	1h RW	**Memory Range Disabled (SIZE_BIT0):** • 1: No TC7 or RS = 1 cycles. • 0: TC7 or RS =1.

20.682 UMA Base Address MSB (UMA_RANGE_LOWER_1)—Offset 384h

This register should be accessed only in ISH3.0. This is an address range register, holds 32 bit MSB value of the address for UMA base range.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**UMA Memory Base Address MSB (UMA_MEMORY_BASE_ADDR_MSB)**

20.683 UMA Limit Address LSB (UMA_RANGE_UPPER_0)—Offset 388h

This register should be accessed only in ISH3.0. This is an address range register, holds 32 bit LSB value of the address for UMA limit range.

Access Method

Type: MEM Register (Size: 32 bits)	**Device:** **Function:**

Default: FFFFF000h

Bit Range	Default & Access	Field Name (ID): Description
31:12	FFFFFh RW	**UMA Memory Limit Address LSB (UMA_MEMORY_LIMIT_ADDR_LSB)**
11:0	0h RO	**UMA Range Upper (SIZE):** NUL UMA Memory size, hard wired to 0 indicate 4 KB window granularity.

20.684 UMA Base Address MSB (UMA_RANGE_UPPER_1)—Offset 38Ch

This register should be accessed only in ISH3.0. This is an address range register, holds 32 bit MSB value of the address for UMA base range.

Access Method

Type: MEM Register (Size: 32 bits)	**Device:** **Function:**

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**UMA Memory Limit Address MSB (UMA_MEMORY_LIMIT_ADDR_MSB)**

20.685 Outbound Inter-processor Messages 1 ISH to ISP (ISH2ISP_MSG1)—Offset 390h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)	**Device:** **Function:**

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.686 Outbound Inter-processor Messages 2 ISH to ISP (ISH2ISP_MSG2)—Offset 394h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.687 Outbound Inter-processor Messages 3 ISH to ISP (ISH2ISP_MSG3)—Offset 398h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.688 Outbound Inter-processor Messages 4 ISH to ISP (ISH2ISP_MSG4)—Offset 39Ch

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.689 Outbound Inter-processor Messages 5 ISH to ISP (ISH2ISP_MSG5)—Offset 3A0h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.690 Outbound Inter-processor Messages 6 ISH to ISP (ISH2ISP_MSG6)—Offset 3A4h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.691 Outbound Inter-processor Messages 7 ISH to ISP (ISH2ISP_MSG7)—Offset 3A8h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.692 Outbound Inter-processor Messages 8 ISH to ISP (ISH2ISP_MSG8)—Offset 3ACh

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.693 Outbound Inter-processor Messages 9 ISH to ISP (ISH2ISP_MSG9)—Offset 3B0h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.694 Outbound Inter-processor Messages 10 ISH to ISP (ISH2ISP_MSG10)—Offset 3B4h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.695 Outbound Inter-processor Messages 11 ISH to ISP (ISH2ISP_MSG11)—Offset 3B8h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.696 Outbound Inter-processor Messages 12 ISH to ISP (ISH2ISP_MSG12)—Offset 3BCh

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.697 Outbound Inter-processor Messages 13 ISH to ISP (ISH2ISP_MSG13)—Offset 3C0h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.698 Outbound Inter-processor Messages 14 ISH to ISP (ISH2ISP_MSG14)—Offset 3C4h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.699 Outbound Inter-processor Messages 15 ISH to ISP (ISH2ISP_MSG15)—Offset 3C8h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.700 Outbound Inter-processor Messages 16 ISH to ISP (ISH2ISP_MSG16)—Offset 3CCh

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.701 Outbound Inter-processor Messages 17 ISH to ISP (ISH2ISP_MSG17)—Offset 3D0h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.702 Outbound Inter-processor Messages 18 ISH to ISP (ISH2ISP_MSG18)—Offset 3D4h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.703 Outbound Inter-processor Messages 19 ISH to ISP (ISH2ISP_MSG19)—Offset 3D8h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.704 Outbound Inter-processor Messages 20 ISH to ISP (ISH2ISP_MSG20)—Offset 3DCh

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.705 Outbound Inter-processor Messages 21 ISH to ISP (ISH2ISP_MSG21)—Offset 3E0h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.706 Outbound Inter-processor Messages 22 ISH to ISP (ISH2ISP_MSG22)—Offset 3E4h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.707 Outbound Inter-processor Messages 23 ISH to ISP (ISH2ISP_MSG23)—Offset 3E8h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.708 Outbound Inter-processor Messages 24 ISH to ISP (ISH2ISP_MSG24)—Offset 3ECh

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.709 Outbound Inter-processor Messages 25 ISH to ISP (ISH2ISP_MSG25)—Offset 3F0h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.710 Outbound Inter-processor Messages 26 ISH to ISP (ISH2ISP_MSG26)—Offset 3F4h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.711 Outbound Inter-processor Messages 27 ISH to ISP (ISH2ISP_MSG27)—Offset 3F8h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.712 Outbound Inter-processor Messages 28 ISH to ISP (ISH2ISP_MSG28)—Offset 3FCh

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.713 Outbound Inter-processor Messages 29 ISH to ISP (ISH2ISP_MSG29)—Offset 400h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.714 Outbound Inter-processor Messages 30 ISH to ISP (ISH2ISP_MSG30)—Offset 404h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.715 Outbound Inter-processor Messages 31 ISH to ISP (ISH2ISP_MSG31)—Offset 408h

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.716 Outbound Inter-processor Messages 32 ISH to ISP (ISH2ISP_MSG32)—Offset 40Ch

Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from ISH to ISP (MSG):** Inter-process message registers for ISH core to communicate to the ISP. These are thirty-two 32 bit registers that hold the message payload from the ISH to ISP. These registers are meant to be written by the ISH and read by ISP.

20.717 Inbound Inter-processor Messages 1 ISP to ISH (ISP2ISH_MSG1)—Offset 410h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.718 Inbound Inter-processor Messages2 ISP to ISH (ISP2ISH_MSG2)—Offset 414h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.719 Inbound Inter-processor Messages 3 ISP to ISH (ISP2ISH_MSG3)—Offset 418h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.720 Inbound Inter-processor Messages 4 ISP to ISH (ISP2ISH_MSG4)—Offset 41Ch

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.721 Inbound Inter-processor Messages 5 ISP to ISH (ISP2ISH_MSG5)—Offset 420h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.722 Inbound Inter-processor Messages 6 ISP to ISH (ISP2ISH_MSG6)—Offset 424h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.723 Inbound Inter-processor Messages 7 ISP to ISH (ISP2ISH_MSG7)—Offset 428h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.724 Inbound Inter-processor Messages 8 ISP to ISH (ISP2ISH_MSG8)—Offset 42Ch

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.725 Inbound Inter-processor Messages 9 ISP to ISH (ISP2ISH_MSG9)—Offset 430h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.726 Inbound Inter-processor Messages 10 ISP to ISH (ISP2ISH_MSG10)—Offset 434h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.727 Inbound Inter-processor Messages 11 ISP to ISH (ISP2ISH_MSG11)—Offset 438h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.728 Inbound Inter-processor Messages 12 ISP to ISH (ISP2ISH_MSG12)—Offset 43Ch

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.729 Inbound Inter-processor Messages 13 ISP to ISH (ISP2ISH_MSG13)—Offset 440h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.730 Inbound Inter-processor Messages 14 ISP to ISH (ISP2ISH_MSG14)—Offset 444h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.731 Inbound Inter-processor Messages 15 ISP to ISH (ISP2ISH_MSG15)—Offset 448h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.732 Inbound Inter-processor Messages 16 ISP to ISH (ISP2ISH_MSG16)—Offset 44Ch

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.733 Inbound Inter-processor Messages 17 ISP to ISH (ISP2ISH_MSG17)—Offset 450h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.734 Inbound Inter-processor Messages 18 ISP to ISH (ISP2ISH_MSG18)—Offset 454h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.735 Inbound Inter-processor Messages 19 ISP to ISH (ISP2ISH_MSG19)—Offset 458h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.736 Inbound Inter-processor Messages 20 ISP to ISH (ISP2ISH_MSG20)—Offset 45Ch

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.737 Inbound Inter-processor Messages 21 ISP to ISH (ISP2ISH_MSG21)—Offset 460h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.738 Inbound Inter-processor Messages 22 ISP to ISH (ISP2ISH_MSG22)—Offset 464h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.739 Inbound Inter-processor Messages 23 ISP to ISH (ISP2ISH_MSG23)—Offset 468h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.740 Inbound Inter-processor Messages 24 ISP to ISH (ISP2ISH_MSG24)—Offset 46Ch

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.741 Inbound Inter-processor Messages 25 ISP to ISH (ISP2ISH_MSG25)—Offset 470h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.742 Inbound Inter-processor Messages 26 ISP to ISH (ISP2ISH_MSG26)—Offset 474h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.743 Inbound Inter-processor Messages 27 ISP to ISH (ISP2ISH_MSG27)—Offset 478h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.744 Inbound Inter-processor Messages 28 ISP to ISH (ISP2ISH_MSG28)—Offset 47Ch

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.745 Inbound Inter-processor Messages 29 ISP to ISH (ISP2ISH_MSG29)—Offset 480h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.746 Inbound Inter-processor Messages 30 ISP to ISH (ISP2ISH_MSG30)—Offset 484h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.747 Inbound Inter-processor Messages 31 ISP to ISH (ISP2ISH_MSG31)—Offset 488h

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.748 Inbound Inter-processor Messages 32 ISP to ISH (ISP2ISH_MSG32)—Offset 48Ch

Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from ISP to ISH (MSG):** Inter-process message registers for ISP to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the ISP to ISH. These registers are meant to be written by the ISP and read by ISH.

20.749 Outbound Inter-processor Messages 1 ISH to Audio (ISH2AUDIO_MSG1)—Offset 490h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.750 Outbound Inter-processor Messages 2 ISH to Audio (ISH2AUDIO_MSG2)—Offset 494h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.751 Outbound Inter-processor Messages 3 ISH to Audio (ISH2AUDIO_MSG3)—Offset 498h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.752 Outbound Inter-processor Messages 4 ISH to Audio (ISH2AUDIO_MSG4)—Offset 49Ch

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.753 Outbound Inter-processor Messages 5 ISH to Audio (ISH2AUDIO_MSG5)—Offset 4A0h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.754 Outbound Inter-processor Messages 6 ISH to Audio (ISH2AUDIO_MSG6)—Offset 4A4h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.755 Outbound Inter-processor Messages 7 ISH to Audio (ISH2AUDIO_MSG7)—Offset 4A8h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.756 Outbound Inter-processor Messages 8 ISH to Audio (ISH2AUDIO_MSG8)—Offset 4ACh

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.757 Outbound Inter-processor Messages 9 ISH to Audio (ISH2AUDIO_MSG9)—Offset 4B0h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.758 Outbound Inter-processor Messages 10 ISH to Audio (ISH2AUDIO_MSG10)—Offset 4B4h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.759 Outbound Inter-processor Messages 11 ISH to Audio (ISH2AUDIO_MSG11)—Offset 4B8h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.760 Outbound Inter-processor Messages 12 ISH to Audio (ISH2AUDIO_MSG12)—Offset 4BCh

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.761 Outbound Inter-processor Messages 13 ISH to Audio (ISH2AUDIO_MSG13)—Offset 4C0h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.762 Outbound Inter-processor Messages 14 ISH to Audio (ISH2AUDIO_MSG14)—Offset 4C4h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.763 Outbound Inter-processor Messages 15 ISH to Audio (ISH2AUDIO_MSG15)—Offset 4C8h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.764 Outbound Inter-processor Messages 16 ISH to Audio (ISH2AUDIO_MSG16)—Offset 4CCh

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.765 Outbound Inter-processor Messages 17 ISH to Audio (ISH2AUDIO_MSG17)—Offset 4D0h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.766 Outbound Inter-processor Messages 18 ISH to Audio (ISH2AUDIO_MSG18)—Offset 4D4h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.767 Outbound Inter-processor Messages 19 ISH to Audio (ISH2AUDIO_MSG19)—Offset 4D8h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.768 Outbound Inter-processor Messages 20 ISH to Audio (ISH2AUDIO_MSG20)—Offset 4DCh

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.769 Outbound Inter-processor Messages 21 ISH to Audio (ISH2AUDIO_MSG21)—Offset 4E0h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.770 Outbound Inter-processor Messages 22 ISH to Audio (ISH2AUDIO_MSG22)—Offset 4E4h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.771 Outbound Inter-processor Messages 23 ISH to Audio (ISH2AUDIO_MSG23)—Offset 4E8h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.772 Outbound Inter-processor Messages 24 ISH to Audio (ISH2AUDIO_MSG24)—Offset 4ECh

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.773 Outbound Inter-processor Messages 25 ISH to Audio (ISH2AUDIO_MSG25)—Offset 4F0h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.774 Outbound Inter-processor Messages 26 ISH to Audio (ISH2AUDIO_MSG26)—Offset 4F4h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.775 Outbound Inter-processor Messages 27 ISH to Audio (ISH2AUDIO_MSG27)—Offset 4F8h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.776 Outbound Inter-processor Messages 28 ISH to Audio (ISH2AUDIO_MSG28)—Offset 4FCh

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.777 Outbound Inter-processor Messages 29 ISH to Audio (ISH2AUDIO_MSG29)—Offset 500h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.778 Outbound Inter-processor Messages 30 ISH to Audio (ISH2AUDIO_MSG30)—Offset 504h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.779 Outbound Inter-processor Messages 31 ISH to Audio (ISH2AUDIO_MSG31)—Offset 508h

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.780 Outbound Inter-processor Messages 32 ISH to Audio (ISH2AUDIO_MSG32)—Offset 50Ch

Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from ISH to Audio (MSG):** Inter-process message registers for ISH core to communicate to the audio. These are thirty-two 32 bit registers that hold the message payload from the ISH to audio. These registers are meant to be written by the ISH and read by audio.

20.781 Inbound Inter-processor Messages 1 Audio to ISH (AUDIO2ISH_MSG1)—Offset 510h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.782 Inbound Inter-processor Messages 2 Audio to ISH (AUDIO2ISH_MSG2)—Offset 514h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.783 Inbound Inter-processor Messages 3 Audio to ISH (AUDIO2ISH_MSG3)—Offset 518h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.784 Inbound Inter-processor Messages 4 Audio to ISH (AUDIO2ISH_MSG4)—Offset 51Ch

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.785 Inbound Inter-processor Messages 5 Audio to ISH (AUDIO2ISH_MSG5)—Offset 520h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.786 Inbound Inter-processor Messages 6 Audio to ISH (AUDIO2ISH_MSG6)—Offset 524h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.787 Inbound Inter-processor Messages 7 Audio to ISH (AUDIO2ISH_MSG7)—Offset 528h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.788 Inbound Inter-processor Messages 8 Audio to ISH (AUDIO2ISH_MSG8)—Offset 52Ch

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.789 Inbound Inter-processor Messages 9 Audio to ISH (AUDIO2ISH_MSG9)—Offset 530h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.790 Inbound Inter-processor Messages 10 Audio to ISH (AUDIO2ISH_MSG10)—Offset 534h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.791 Inbound Inter-processor Messages 11 Audio to ISH (AUDIO2ISH_MSG11)—Offset 538h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.792 Inbound Inter-processor Messages 12 Audio to ISH (AUDIO2ISH_MSG12)—Offset 53Ch

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.793 Inbound Inter-processor Messages 13 Audio to ISH (AUDIO2ISH_MSG13)—Offset 540h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.794 Inbound Inter-processor Messages 14 Audio to ISH (AUDIO2ISH_MSG14)—Offset 544h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.795 Inbound Inter-processor Messages 15 Audio to ISH (AUDIO2ISH_MSG15)—Offset 548h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.796 Inbound Inter-processor Messages 16 Audio to ISH (AUDIO2ISH_MSG16)—Offset 54Ch

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.797 Inbound Inter-processor Messages 17 Audio to ISH (AUDIO2ISH_MSG17)—Offset 550h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.798 Inbound Inter-processor Messages 18 Audio to ISH (AUDIO2ISH_MSG18)—Offset 554h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.799 Inbound Inter-processor Messages 19 Audio to ISH (AUDIO2ISH_MSG19)—Offset 558h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.800 Inbound Inter-processor Messages 20 Audio to ISH (AUDIO2ISH_MSG20)—Offset 55Ch

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.801 Inbound Inter-processor Messages 21 Audio to ISH (AUDIO2ISH_MSG21)—Offset 560h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.802 Inbound Inter-processor Messages 22 Audio to ISH (AUDIO2ISH_MSG22)—Offset 564h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.803 Inbound Inter-processor Messages 23 Audio to ISH (AUDIO2ISH_MSG23)—Offset 568h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.804 Inbound Inter-processor Messages 24 Audio to ISH (AUDIO2ISH_MSG24)—Offset 56Ch

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.805 Inbound Inter-processor Messages 25 Audio to ISH (AUDIO2ISH_MSG25)—Offset 570h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.806 Inbound Inter-processor Messages 26 Audio to ISH (AUDIO2ISH_MSG26)—Offset 574h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.807 Inbound Inter-processor Messages 27 Audio to ISH (AUDIO2ISH_MSG27)—Offset 578h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.808 Inbound Inter-processor Messages 28 Audio to ISH (AUDIO2ISH_MSG28)—Offset 57Ch

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.809 Inbound Inter-processor Messages 29 Audio to ISH (AUDIO2ISH_MSG29)—Offset 580h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.810 Inbound Inter-processor Messages 30 Audio to ISH (AUDIO2ISH_MSG30)—Offset 584h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.811 Inbound Inter-processor Messages 31 Audio to ISH (AUDIO2ISH_MSG31)—Offset 588h

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.812 Inbound Inter-processor Messages 32 Audio to ISH (AUDIO2ISH_MSG32)—Offset 58Ch

Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from Audio to ISH (MSG):** Inter-process message registers for audio to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the audio to ISH. These registers are meant to be written by the audio and read by ISH.

20.813 Outbound Inter-processor Messages 1 ISH to GFX (ISH2GFX_MSG1)—Offset 590h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.814 Outbound Inter-processor Messages 2 ISH to GFX (ISH2GFX_MSG2)—Offset 594h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.815 Outbound Inter-processor Messages 3 ISH to GFX (ISH2GFX_MSG3)—Offset 598h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.816 Outbound Inter-processor Messages 4 ISH to GFX (ISH2GFX_MSG4)—Offset 59Ch

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.817 Outbound Inter-processor Messages 5 ISH to GFX (ISH2GFX_MSG5)—Offset 5A0h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.818 Outbound Inter-processor Messages 6 ISH to GFX (ISH2GFX_MSG6)—Offset 5A4h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.819 Outbound Inter-processor Messages 7 ISH to GFX (ISH2GFX_MSG7)—Offset 5A8h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.820 Outbound Inter-processor Messages 8 ISH to GFX (ISH2GFX_MSG8)—Offset 5ACh

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to Gfx. These registers are meant to be written by the ISH and read by GFX.

20.821 Outbound Inter-processor Messages 9 ISH to GFX (ISH2GFX_MSG9)—Offset 5B0h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.822 Outbound Inter-processor Messages 10 ISH to GFX (ISH2GFX_MSG10)—Offset 5B4h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.823 Outbound Inter-processor Messages 11 ISH to GFX (ISH2GFX_MSG11)—Offset 5B8h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.824 Outbound Inter-processor Messages 12 ISH to GFX (ISH2GFX_MSG12)—Offset 5BCh

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.825 Outbound Inter-processor Messages 13 ISH to GFX (ISH2GFX_MSG13)—Offset 5C0h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.826 Outbound Inter-processor Messages 14 ISH to GFX (ISH2GFX_MSG14)—Offset 5C4h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.827 Outbound Inter-processor Messages 15 ISH to GFX (ISH2GFX_MSG15)—Offset 5C8h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.828 Outbound Inter-processor Messages 16 ISH to GFX (ISH2GFX_MSG16)—Offset 5CCh

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.829 Outbound Inter-processor Messages 17 ISH to GFX (ISH2GFX_MSG17)—Offset 5D0h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.830 Outbound Inter-processor Messages 18 ISH to GFX (ISH2GFX_MSG18)—Offset 5D4h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.831 Outbound Inter-processor Messages 19 ISH to GFX (ISH2GFX_MSG19)—Offset 5D8h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.832 Outbound Inter-processor Messages 20 ISH to GFX (ISH2GFX_MSG20)—Offset 5DCh

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.833 Outbound Inter-processor Messages 21 ISH to GFX (ISH2GFX_MSG21)—Offset 5E0h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.834 Outbound Inter-processor Messages 22 ISH to GFX (ISH2GFX_MSG22)—Offset 5E4h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.835 Outbound Inter-processor Messages 23 ISH to GFX (ISH2GFX_MSG23)—Offset 5E8h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.836 Outbound Inter-processor Messages 24 ISH to GFX (ISH2GFX_MSG24)—Offset 5ECh

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.837 Outbound Inter-processor Messages 25 ISH to GFX (ISH2GFX_MSG25)—Offset 5F0h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.838 Outbound Inter-processor Messages 26 ISH to GFX (ISH2GFX_MSG26)—Offset 5F4h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.839 Outbound Inter-processor Messages 27 ISH to GFX (ISH2GFX_MSG27)—Offset 5F8h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.840 Outbound Inter-processor Messages 28 ISH to GFX (ISH2GFX_MSG28)—Offset 5FCh

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.841 Outbound Inter-processor Messages 29 ISH to GFX (ISH2GFX_MSG29)—Offset 600h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.842 Outbound Inter-processor Messages 30 ISH to GFX (ISH2GFX_MSG30)—Offset 604h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.843 Outbound Inter-processor Messages 31 ISH to GFX (ISH2GFX_MSG31)—Offset 608h

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.844 Outbound Inter-processor Messages 32 ISH to GFX (ISH2GFX_MSG32)—Offset 60Ch

Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from ISH to GFX (MSG):** Inter-process message registers for ISH core to communicate to the GFX. These are thirty-two 32 bit registers that hold the message payload from the ISH to GFX. These registers are meant to be written by the ISH and read by GFX.

20.845 Inbound Inter-processor Messages 1 GFX to ISH (GFX2ISH_MSG1)—Offset 610h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 1 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.846 Inbound Inter-processor Messages 2 GFX to ISH (GFX2ISH_MSG2)—Offset 614h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 2 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.847 Inbound Inter-processor Messages 3 GFX to ISH (GFX2ISH_MSG3)—Offset 618h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 3 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.848 Inbound Inter-processor Messages 4 GFX to ISH (GFX2ISH_MSG4)—Offset 61Ch

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 4 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.849 Inbound Inter-processor Messages 5 GFX to ISH (GFX2ISH_MSG5)—Offset 620h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 5 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.850 Inbound Inter-processor Messages 6 GFX to ISH (GFX2ISH_MSG6)—Offset 624h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 6 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.851 Inbound Inter-processor Messages 7 GFX to ISH (GFX2ISH_MSG7)—Offset 628h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 7 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.852 Inbound Inter-processor Messages 8 GFX to ISH (GFX2ISH_MSG8)—Offset 62Ch

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 8 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.853 Inbound Inter-processor Messages 9 GFX to ISH (GFX2ISH_MSG9)—Offset 630h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 9 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.854 Inbound Inter-processor Messages 10 GFX to ISH (GFX2ISH_MSG10)—Offset 634h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 10 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.855 Inbound Inter-processor Messages 11 GFX to ISH (GFX2ISH_MSG11)—Offset 638h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 11 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.856 Inbound Inter-processor Messages 12 GFX to ISH (GFX2ISH_MSG12)—Offset 63Ch

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 12 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.857 Inbound Inter-processor Messages 13 GFX to ISH (GFX2ISH_MSG13)—Offset 640h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 13 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.858 Inbound Inter-processor Messages 14 GFX to ISH (GFX2ISH_MSG14)—Offset 644h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 14 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.859 Inbound Inter-processor Messages 15 GFX to ISH (GFX2ISH_MSG15)—Offset 648h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 15 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.860 Inbound Inter-processor Messages 16 GFX to ISH (GFX2ISH_MSG16)—Offset 64Ch

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 16 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.861 Inbound Inter-processor Messages 17 GFX to ISH (GFX2ISH_MSG17)—Offset 650h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 17 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.862 Inbound Inter-processor Messages 18 GFX to ISH (GFX2ISH_MSG18)—Offset 654h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 18 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.863 Inbound Inter-processor Messages 19 GFX to ISH (GFX2ISH_MSG19)—Offset 658h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 19 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.864 Inbound Inter-processor Messages 20 GFX to ISH (GFX2ISH_MSG20)—Offset 65Ch

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 20 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.865 Inbound Inter-processor Messages 21 GFX to ISH (GFX2ISH_MSG21)—Offset 660h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 21 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.866 Inbound Inter-processor Messages 22 GFX to ISH (GFX2ISH_MSG22)—Offset 664h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 22 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.867 Inbound Inter-processor Messages 23 GFX to ISH (GFX2ISH_MSG23)—Offset 668h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 23 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.868 Inbound Inter-processor Messages 24 GFX to ISH (GFX2ISH_MSG24)—Offset 66Ch

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 24 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.869 Inbound Inter-processor Messages 25 GFX to ISH (GFX2ISH_MSG25)—Offset 670h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 25 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.870 Inbound Inter-processor Messages 26 GFX to ISH (GFX2ISH_MSG26)—Offset 674h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 26 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.871 Inbound Inter-processor Messages 27 Gfx to ISH (GFX2ISH_MSG27)—Offset 678h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 27 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.872 Inbound Inter-processor Messages 28 GFX to ISH (GFX2ISH_MSG28)—Offset 67Ch

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 28 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.873 Inbound Inter-processor Messages 29 GFX to ISH (GFX2ISH_MSG29)—Offset 680h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 29 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.874 Inbound Inter-processor Messages 30 GFX to ISH (GFX2ISH_MSG30)—Offset 684h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 30 from GFX to ISH (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.875 Inbound Inter-processor Messages 31 GFX to ISH (GFX2ISH_MSG31)—Offset 688h

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 31 from GFX to ISH. (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.876 Inbound Inter-processor Messages 32 GFX to ISH (GFX2ISH_MSG32)—Offset 68Ch

Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Message 32 from GFX to ISH. (MSG):** Inter-process message registers for GFX to communicate to the ISH. These are thirty-two 32 bit registers that hold the message payload from the GFX to ISH. These registers are meant to be written by the GFX and read by ISH.

20.877 Inbound Doorbell ISP to ISH (ISP2ISH_DOORBELL)—Offset 690h

Inbound doorbell register, ISP core to interrupt ISH. Data 30:0 is 31 bit message payload used for backward compatibility. Software can use either this 31 bit message payload or 256 bit payload registers.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** ISP to ISH doorbell busy bit. When this bit is cleared, the ISH CPU is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** ISP to ISH doorbell 31 bit payload, 31 bit message payload for backward compatibility.

20.878 Outbound Doorbell ISH to ISP (ISH2ISP_DOORBELL)—Offset 694h

Outbound doorbell register for the ISH to interrupt the ISP. Setting bit 31 of this register causes the ISP to receive a legacy INTA interrupt. Data 30:0 is 31 bit message payload used for backward compatibility.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** ISH to ISP doorbell busy bit. When this bit is cleared, the ISP is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** ISH to ISP doorbell 31 bit payload, 31 bit message payload for backward compatibility.

20.879 Inbound Doorbell Audio to ISH (AUDIO2ISH_DOORBELL)—Offset 698h

Inbound doorbell register, audio core to interrupt ISH. Data 30:0 is 31 bit message payload used for backward compatibility. Software can use either this 31 bit message payload or 256 bit payload registers.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY)**
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** Audio to ISH doorbell 31 bit payload, 31 bit message payload for backward compatibility.

20.880 Outbound Doorbell ISH to Audio (ISH2AUDIO_DOORBELL)—Offset 69Ch

Outbound doorbell register for the ISH to interrupt the audio. Setting bit 31 of this register causes the audio to receive a legacy INTA interrupt. Data 30:0 is 31 bit message payload used for backward compatibility. Software can use either this 31 bit message payload or 256 bit payload registers.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** ISH to audio doorbell busy bit. When this bit is cleared, the audio is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** ISH to audio doorbell 31 bit payload, 31 bit message payload for backward compatibility.

20.881 Inbound Doorbell GFX to ISH (GFX2ISH_DOORBELL)—Offset 6A0h

Inbound doorbell register, GFX core to interrupt ISH. Data 30:0 is 31 bit message payload used for backward compatibility. Software can use either this 31 bit message payload or 256 bit payload registers.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** GFX to ISH doorbell busy bit. When this bit is cleared, the ISH CPU is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** GFX to ISH doorbell 31 bit payload, 31 bit message payload for backward compatibility.

20.882 Outbound Doorbell ISH to GFX (ISH2GFX_DOORBELL)—Offset 6A4h

Outbound doorbell register for the ISH to interrupt the GFX. Setting bit 31 of this register causes the GFX to receive a legacy INTA interrupt. Data 30:0 is 31 bit message payload used for backward compatibility. Software can use either this 31 bit message payload or 256 bit payload registers.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Doorbell Busy Bit (BUSY):** ISH to GFX doorbell busy bit. When this bit is cleared, the GFX is ready to accept a new message.
30:0	0h RW	**31 Bit Payload (PAYLOAD_31BIT):** ISH to GFX doorbell 31 bit payload, 31 bit message payload for backward compatibility.

20.883 CSME to ISH Control and Status (CSME2ISH_CTRL_STATUS)—Offset 6A8h

This is a 32 bit register which is set to 0 on a system reset or a wakeup from D3Cold. When CSME writes to any of these bits with 1, an interrupt is genrated to mIA. The Interrupt is then cleared by mIA by writing 1`b1 to this register after the appropriate ISR is serviced.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/1S	**32 Bit Payload (CSME2ISH_CTRL_STATUS):** Control and status information transfer from CSME FW to ISH FW.

20.884 ISH IPC D0I3 Control (IPC_d0i3C_reg)—Offset 6D0h

This is a 32 bit register which is set to 0 on a system reset or a wakeup from D3Cold. When CSME writes to any of these bits with 1, an interrupt is genrated to mIA. The Interrupt is then cleared by mIA by writing 1`b1 to this register after the appropriate ISR is serviced.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 8h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RO	**Reserved (RESERVED):** Reserved.
4	0h RO	**Interrupt Request Capable (ISH_IPC_d0i3C_reg_RSVD)**
3	1h RW/1C	**Restore Required (ISH_IPC_d0i3C_reg3):** D0i3 register for storing power related information RR: Restore Required bit.
2	0h RW	**D0i3 (ISH_IPC_d0i3C_reg2):** D0i3 register for storing power related information i3: D0i3 bit.
1	0h RW	**Interrupt Request (ISH_IPC_d0i3C_reg1):** D0i3 register for storing power related information IR: Interrupt Request bit.
0	0h RW/1C	**Command-in-Progress (ISH_IPC_d0i3C_reg0):** D0i3 register for storing power related information CIP: Command-in-Progress bit.

20.885 PMC to ISH Control and Status (ipc_pmc2ish_csr_reg)—Offset 6D4h

This is a 32 bit register which is set to 0 on a system reset or a wakeup from D3Cold. When PMC writes to any of these bits with 1, an interrupt is generated to mIA. The Interrupt is then cleared by mIA by writing 1`b1 to this register after the appropriate ISR is serviced.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/1S	**32 Bit Payload (ISH_IPC_PMC2ISH_CSR_REG):** Control and status information transfer between ISH and PMC FWs and interrupt generation to mIA.

20.886 ISP to ISH Control and Status (ipc_isp2ish_csr_reg)—Offset 6D8h

This is a control and status register for enabling ISP to interrupt mIA incase of ISP writes to any of these bits of the register. The Interrupt is then cleared by mIA by writing 1`b1 to this register after the appropriate ISR is serviced.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/1S	**32 Bit Payload (ISH_IPC_ISP2ISH_CSR_REG):** Control and status information transfer between ISH and ISP FWs and interrupt generation to mIA.

20.887 Audio to ISH Control and Status (ipc_audio2ish_csr_reg)—Offset 6DCh

This is a control and status register for enabling audio to interrupt mIA incase of audio writes to any of these bits of the register. The Interrupt is then cleared by mIA by writing 1`b1 to this register after the appropriate ISR is serviced.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/1S	**32 Bit Payload (ISH_IPC_AUDIO2ISH_CSR_REG):** Control and status information transfer between ISH and audio FWs and interrupt generation to mIA.

20.888 RID - Bridge revision ID Register (RID)—Offset 30h

register contains the Bridge Revision ID

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved0:** Reserved
7:0	0h RO	**RID:** Revision ID of the Bridge.

20.889 GEN_REGRW1 - General Purpose register (GEN_REGRW1)—Offset 600h

General Purpose PRV RW Register 1.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW1:** This register value is brought out as oob_gen_prv_rw_reg1 out of band signal

20.890 GEN_REGRW2 - General Purpose register (GEN_REGRW2)—Offset 604h

General Purpose PRV RW Register 2.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW2:** This register value is brought out as oob_gen_prv_rw_reg2 out of band signal

20.891 GEN_REGRW3 - General Purpose register (GEN_REGRW3)—Offset 608h

General Purpose PRV RW Register 3.

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW3:** This register value is brought out as oob_gen_prv_rw_reg3 out of band signal

20.892 GEN_REGRW4 - General Purpose register (GEN_REGRW4)—Offset 60Ch

General Purpose PRV RW Register 4.

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REG_RW4:** This register value is brought out as oob_gen_prv_rw_reg4 out of band signal

§ §

21 HECI Registers

This chapter documents the registers of the HECI devices. The processor contains multiple HECI devices:

- Bus: 0, Device: 15, Function: 0 (HECI1)
- Bus: 0, Device: 15, Function: 1 (HECI2)
- Bus: 0, Device: 15, Function: 2 (HECI3)

NOTE: Register default values are taken from device HECI1 only. Refer Vol1 for Device IDs

21.1 Identifiers (HECI1_ID) — Offset 0h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 0h	8086 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO/V	**Device ID (DID):** Indicates device number assigned by Intel. The upper 9 bits of this field are set by a special sideband message (SetIDValue). The lower 7 bits are set by soft straps. Note that the default value of this field is not 0, but depends on the value of soft straps. Refer to CSE spec in order to find the value of 7 LSBs.
15:0	8086h RO	**Vendor ID (VID):** 16-bit field which indicates Intel is the vendor, assigned by the PCI SIG.

21.2 Command (HECI1_CMD) — Offset 4h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + 4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:11	0h RO	**Reserved (RSVD):** Reserved
10	0h RW	**Interrupt Disable (ID):** Disables this device from generating PCI line based interrupts. This bit does not have any effect on MSI operation.
9	0h RO	**Fast Back-to-Back Enable (FBE):** Not implemented, hardwired to 0.
8	0h RO	**SERR# Enable (SEE):** Not implemented, hardwired to 0.
7	0h RO	**Wait Cycle Enable (WCC):** Not implemented, hardwired to 0.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RO	**Parity Error Response Enable (PEE):** Not implemented, hardwired to 0.
5	0h RO	**VGA Palette Snooping Enable (VAG):** Not implemented, hardwired to 0
4	0h RO	**Memory Write And Invalidate Enable (MWIE):** Not implemented, hardwired to 0.
3	0h RO	**Special Cycle Enable (SCE):** Not implemented, hardwired to 0.
2	0h RW	**Bus Master Enable (BME):** Controls the HECI host controller's abiity to act as a system memory master for data transfers. When this bit is cleared, HECI bus master activity stops and any active DMA engines return to an idle condition. This bit is made visible to firmware through the H_PCI_CSR register, and changes to this bit may be configured by the H_PCI_CSR register to generate an MSI to mIA. When this bit is '0', HECI is blocked from generating MSI to the host CPU. Note that this bit does not block HECI accesses to CSE-UMA.
1	0h RW	**Memory Space Enable (MSE):** Controls access to the HECI host controller's memory mapped register space.
0	0h RO	**I/O Space Enable (IOSE):** Not implemented, hardwired to 0.

21.3 Status (HECI1_STS) — Offset 6h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + 6h	10 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**Detected Parity Error (DPE):** Not implemented, hardwired to 0.
14	0h RO	**Signaled System Error (SSE):** Not implemented, hardwired to 0.
13	0h RO	**Received Master-Abort (RMA):** Not implemented, hardwired to 0.
12	0h RO	**Received Target Abort (RTA):** Not implemented, hardwired to 0.
11	0h RO	**Signaled Target-Abort (STA):** Not implemented, hardwired to 0.
10:9	0h RO	**DEVSEL# Timing (DEVT):** These bits are hardwired to 00.
8	0h RO	**Master Data Pariy Error Detected (DPD):** Not implemented, hardwired to 0.
7	0h RO	**Fast Back-to-Back Capable (FBC):** Not implemented, hardwired to 0.
6	0h RO	**Reserved (RSVD_6_6):** Reserved
5	0h RO	**66 MHz Capable (C66):** Not implemented, hardwired to 0.

Bit Range	Default & Access	Field Name (ID): Description
4	1h RO	**Capabilities List (CL):** Indicates the presence of a capabilities list, hardwired to 1.
3	0h RO/V	**Interrupt Status (IS):** Reflects the state of the INTx# signal at the input of the enable/disable circuit. This bit is a 1 when the INTx# is asserted. This bit is a 0 after the interrupt is cleared (independent of the state of the Interrupt Disable bit in the command register). Note that this bit is not set by an MSI.
2:0	0h RO	**Reserved (RSVD_2_0):** Reserved

21.4 Revision ID And Class Code (HECI1_RID_CC) — Offset 8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 8h	7800000 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	7h RO	**Base Class Code (BCC):** Indicates the base class code of the HECI host controller device.
23:16	80h RO	**Sub Class Code (SCC):** Indicates the sub class code of the HECI host controller device.
15:8	0h RO	**Programming Interface (PI):** Indicates the programming interface of the HECI host controller device.
7:0	0h RO/V	**Revision ID (RID):** Indicates stepping of the HECI host controller. This field is set by a special IOSF SB message (SetIDValue).

21.5 Cache Line Size (HECI1_CLS) — Offset Ch

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:0] + Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Cache Line Size (CLS):** Not implemented, hardwired to 0.

21.6 Master Latency Timer (HECI1_MLT) — Offset Dh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:0] + Dh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Master Latency Timer (MLT):** Not implemented, hardwired to 0.

21.7 Header Type (HECI1_HTYPE) — Offset Eh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:0] + Eh	80 h

Bit Range	Default & Access	Field Name (ID): Description
7	1h RO	**Multi-Function Device (MFD):** Indicates the HECI host controller is part of a multi-function device.
6:0	0h RO	**Header Layout (HL):** Indicates that the HECI host controller uses a target device layout.

21.8 Built In Self-Test (HECI1_BIST) — Offset Fh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:0] + Fh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**BIST Capable (BC):** Not implemented, hardwired to 0.
6:0	0h RO	**Reserved (RSVD):** Reserved

21.9 HECI MMIO Base Address Low (HECI1_MMIO_MBAR_LO) — Offset 10h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 10h	4 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**Base Address Low (BA_LO):** Lower 32 bits of base address of register memory space.
11:4	0h RO	**Memory Size (MS):** This field is hardwired to 0 to indicate that HECI device requests 4KB of memory space.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO	**Prefetchable (PF):** Indicates that this range is not pre-fetchable.
2:1	2h RO	**Type (TP):** Indicates that this range can be mapped anywhere in 64-bit address space.
0	0h RO	**Resource Type Indicator (RTE):** Indicates a request for register memory space.

21.10 HECI MMIO Base Address High (HECI1_MMIO_MBAR_HI) — Offset 14h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Base Address High (BA_HI):** Upper 32 bits of base address of register memory space.

21.11 Sub System Identifiers (HECI1_SS) — Offset 2Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**Subsystem ID (SSID):** Indicates the subsystem identifier. This field should be programmed by BIOS during boot-up. Once written, this register becomes Read Only.
15:0	0h RW/O	**Subsystem Vendor ID (SSVID):** Indicates the subsystem vendor identifier. This field should be programmed by BIOS during boot- up. Once written, this register becomes Read Only.

21.12 Capabilities Pointer (HECI1_CAP) — Offset 34h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 34h	50 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
7:0	50h RO	**Capability Pointer (CP):** Indicates the first capability pointer offset. It points to the PCI power management capability offset.

21.13 Interrupt Information (HECI1_INTR) — Offset 3Ch

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + 3Ch	100 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	1h RO/V	**Interrupt Pin (IPIN):** This field indicates the virtual interrupt pin the HECI host controller uses. A value of 0x1/0x2/0x3/0x4 indicates that this function implements legacy interrupt on INTA/INTB/INTC/INTD, respectively. When legacy interrupt is not used this register should be set to 0x0. Programmed by BIOS over IOSF SB through corresponding Private CR register.
7:0	0h RW	**Interrupt Line (ILINE):** Software written value to indicate which interrupt line (vector) the interrupt is connected to. No hardware action is taken on this field.

21.14 Minimum Grant (HECI1_MGNT) — Offset 3Eh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:0] + 3Eh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Grant (GNT):** Not implemented, hardwired to 0.

21.15 Maximum Latency (HECI1_MLAT) — Offset 3Fh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:0] + 3Fh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**LAT:** Not implemented, hardwired to 0.

21.16 Host Firmware Status (HECI1_HFS) — Offset 40h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 40h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Firmware Status Host Access (FS_HA):** Indicates current status of the firmware for the HECI controller. This field is the host's read only access to the FS field in the CSE Firmware Status register.

21.17 Miscellaneous Shadow (HECI1_MISC_SHDW) — Offset 44h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 44h	80000000 h

Bit Range	Default & Access	Field Name (ID): Description
31	1h RO	**Miscellaneous Shadow Valid (MSVLD):** This bit is hardwired to 1 to indicate that this HECI device implements the Miscellaneous Shadow register. This bit can be used by host software that is bus/dev/function number agnostic (such as HECI operating system driver) to discover whether the Miscellaneous Shadow register is implemented or not.
30:17	0h RO	**Reserved (RSVD_30_17):** Reserved
16	0h RO/V	**CSE UMA Size Valid (CUSZV):** This bit indicates that FW has written the CUSZ field. This field reflects the value of CSE CUBA.CUSZV.
15:7	0h RO	**Reserved (RSVD_15_7):** Reserved
6:0	0h RO/V	**CSE UMA Size (CUSZ):** These bits reflect firmware's desired size of CSEUMA memory region. It is configured by firmware prior to bring up core power and allowing BIOS to initialize memory. The legal CSEUMA sizes and encodings for this field are: 0000000b (0MB, no CSEUMA region) 0000001b (1MB) 0000010b (2MB) 0000100b (4MB) 0001000b (8MB) 0010000b (16MB) 0100000b (32MB) 1000000b (64MB). This field reflects the value of CSE CUBA.CUSZ.

21.18 General Status Shadow 1 (HECI1_GS_SHDW1) — Offset 48h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 48h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 1 (GSS1):** This field is host side shadow of CSE General Status 1 (CSE_GS1).

21.19 Host General Status (HECI1_H_GS1) — Offset 4Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 4Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Host General Status (H_GS1):** General status of Host. This field is not used by hardware.

21.20 PCI Power Management Capability ID (HECI1_PID) — Offset 50h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + 50h	8C01 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	8Ch RO	**NEXT:** Indicates the location of the next capability item in the list. This is the Message Signaled Interrupts capability.
7:0	1h RO	**Cap ID (CID):** Indicates that this pointer is a PCI power management.

21.21 PCI Power Management Capabilities (HECI1_PC) — Offset 52h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + 52h	4003 h

Bit Range	Default & Access	Field Name (ID): Description
15:11	8h RO	**PME Support (PSUP):** Indicates the states that can generate PME#. HECI can assert PME# from D3hot only.
10	0h RO	**D2 Support (D2S):** The D2 state is not supported for the HECI host controller.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RO	**D1 Support (D1S):** The D1 state is not supported for the HECI host controller.
8:6	0h RO	**Aux Current (AUXC):** Reports the maximum Suspend well current required when in the D3COLD state. Value of 0 is reported.
5	0h RO	**Device Specific Initialization (DSI):** Indicates whether device-specific initialization is required.
4	0h RO	**Reserved (RSVD):** Reserved
3	0h RO	**PME Clock (PMEC):** Indicates that PCI clock is not required to generate PME#.
2:0	3h RO	**Version (VS):** Indicates support for Revision 1.2 of the PCI Power Management Specification.

21.22 PCI Power Management Control And Status (HECI1_PMCS) — Offset 54h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + 54h	8 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C/V	**PME Status (PMES):** The PME Status bit in HECI space can be set to '1' by CSE FW performing a write into CSE side register to set PMES. This bit is cleared by host CPU writing a '1' to it. CSE cannot clear this bit. Host CPU writes with value '0' have no effect on this bit.
14:9	0h RO	**Reserved (RSVD_14_9):** Reserved
8	0h RW	**PME Enable (PMEE):** When set, PME_assert and PME_deassert messages are sent over Sideband to PMC based on the PMES bit. When reset these messages may not be sent.
7:4	0h RO	**Reserved (RSVD_7_4):** Reserved
3	1h RO	**No Soft Reset (NSR):** This bit indicates that when the HECI host controller is transitioning from D3hot to D0 due to power state command, it does not perform an internal reset.
2	0h RO	**Reserved (RSVD_2_2):** Reserved
1:0	0h RW	**Power State (PS):** This field is used both to determine the current power state of the HECI host controller and to set a new power state. The values are: 00 - D0 state 11 - D3HOT state. The D1 and D2 states are not supported for this HECI host controller. If software attempts to write an unsupported, optional state to this field, the write operation must complete normally on the bus; however, the data is discarded and no state change occurs. When in the D3HOT state, the HBA's configuration space is available, but the registers memory space is not. Additionally, interrupts are blocked. This field is visible to firmware through the H_PCI_CSR register, and changes to this field may be configured by the H_PCI_CSR register to generate an MSI to mIA.

21.23 General Status Shadow 2 (HECI1_GS_SHDW2) — Offset 60h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 60h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 2 (GSS2):** This field is host side shadow of CSE General Status 2 (CSE_GS2).

21.24 General Status Shadow 3 (HECI1_GS_SHDW3) — Offset 64h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 64h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 3 (GSS3):** This field is host side shadow of CSE General Status 3 (CSE_GS3).

21.25 General Status Shadow 4 (HECI1_GS_SHDW4) — Offset 68h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 68h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 4 (GSS4):** This field is host side shadow of CSE General Status 4 (CSE_GS4).

21.26 General Status Shadow 5 (HECI1_GS_SHDW5) — Offset 6Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 6Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 5 (GSS5):** This field is host side shadow of CSE General Status 5 (CSE_GS5).

21.27 Host General Status 2 (HECI1_H_GS2) — Offset 70h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 70h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Host General Status 2 (H_GS2):** General status of Host. This field is not used by hardware.

21.28 Host General Status 3 (HECI1_H_GS3) — Offset 74h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 74h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Host General Status 3 (H_GS3):** General status of Host. This field is not used by hardware.

21.29 Message Signaled Interrupt Identifiers (HECI1_MID) — Offset 8Ch

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + 8Ch	A405 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	A4h RO	**NEXT:** Indicates the location of the next capability item in the list. This is the DevIdle conventional PCI vendor specific capability.
7:0	5h RO	**Capability ID (CID):** Indicates MSI.

21.30 Message Signaled Interrupt Message Control (HECI1_MC) — Offset 8Eh

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + 8Eh	80 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	**Reserved (RSVD):** Reserved
7	1h RO	**64 Bit Address Capable (C64):** Specifies whether capable of generating 64-bit messages.
6:4	0h RO	**Multiple Message Enable (MME):** Not implemented, hardwired to 0.
3:1	0h RO	**Multiple Message Capable (MMC):** Not implemented, hardwired to 0.
0	0h RW	**MSI Enable (MSIE):** If set, MSI is enabled and INTx virtual wires are not used to generate interrupts.

21.31 Message Signaled Interrupt Message Address (HECI1_MA) — Offset 90h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 90h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**ADDR:** Lower 32 bits of the system specified message address, always DW aligned.
1:0	0h RO	**Reserved (RSVD):** Reserved

21.32 Message Signaled Interrupt Upper Address (HECI1_MUA) — Offset 94h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 94h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Upper Address (UADDR):** Upper 32 bits of the system specified message address. This register is optional and only implemented if MC.C64=1.

21.33 Message Signaled Interrupt Message Data (HECI1_MD) — Offset 98h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + 98h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW	**DATA:** This 16-bit field is programmed by system software if MSI is enabled. Its content is driven onto the FSB during the data phase of the MSI memory write transaction.

21.34 HECI Interrupt Delivery Mode (HECI1_HIDM) — Offset A0h

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:0] + A0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:3	0h RO	**Reserved (RSVD):** Reserved
2	0h RW/1S/V	**HIDM Lock (HIDM_L):** Writing 1 to this bit locks the HIDM field.
1:0	0h RW/L	**HECI Interrupt Delivery Mode (HIDM):** These bits control what type of interrupt the HECI will send when CSE FW writes to set the CSE_IG bit. They are interpreted as follows: 00 - Generate Legacy or MSI interrupt; 01 - Generate SCI; 10 - Generate SMI; This field may be locked by writing 1 to HIDM_L bit.

21.35 Vendor Specific Capability Register (HECI1_VSCR) — Offset A4h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + A4h	F0140009 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RO	**Vendor-Specific Capability ID (VSID):** A value of 4'hF in this 4-bit field identifies this Vendor Specific Capability as an Extended Capability, which uses a VSEC 16-bit Extended Capability ID in the subsequent four bytes. This field allows software to differentiate this capability from other Vendor Specific capabilities.
27:24	0h RO	**Vendor Specific Capability Revision (VSREV):** For a VSID of 4'hF, this field is reserved 4'h0.
23:16	14h RO	**Vendor Specific Capability Length (VSLEN):** This field indicates the number of bytes of this Vendor Specific capability as required by the PCI spec inclusive of the CapID and Next Pointer. It has the value of 8'h14 for the DevIdle Capability.

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	**NEXT:** Pointer to the configuration offset of the next Capability item. It is 8'h00 since it is the final item in the Capability list.
7:0	9h RO	**Capability ID (CAPID):** The value of 8h09 in this field indicates a Vendor Specific capability.

21.36 Vendor Specific Extended Capability Register (HECI1_VSEC) — Offset A8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + A8h	1400010 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	14h RO	**Vendor Specific Extended Capability Length (VSECLEN):** This field indicates the number of bytes of this Vendor Specific capability inclusive of the 1st DW which includes the capability ID and Next pointer. For DevIdle, this is 12'h14.
19:16	0h RO	**Vendor Specific Extended Capability Revision (VSECREV):** For this revision of DevIdle, this field is 4h0.
15:0	10h RO	**Vendor Specific Extended Capability ID (NEXT):** DevIdle has been assigned the Intel VSEC ID of 16h10.

21.37 SW LTR Pointer Register (HECI1_SWLTRPTR) — Offset ACh

Software Latency Tolerance Reporting (LTR) Pointer Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + ACh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**SW LTR Update MMIO Offset Location (SWLTRLOC):** This register contains the location pointer to the SW LTR register in MMIO space, as an offset from the specified BAR. The value of this register is not valid, if the Valid bit is not set.
3:1	0h RO	**Base Address Register Number (BARNUM):** Contains the 0's based AR Number of the BAR which contains the location of the SW LTR MMIO register. When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number. Bar 0 is the 1st BAR. The value of this register is not valid, if the Valid bit is not set.
0	0h RO	**VALID:** Set to 1'b1 to indicate that the function has implemented a SW LTR register as specified in the DevIdle that can be located using the SWLTRLOC register and BARNUM. Set to 1'b0 to indicate that the function has not implemented a SW LTR register that is compliant to the DevIdle definition. This could be because the function has not implemented SW LTR at all, or has a device specific version of SW LTR.

21.38 Device Idle Pointer Register (HECI1_DEVIDLEPTR) — Offset B0h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + B0h	8001 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	800h RO	**Device Idle MMIO Offset Location (DEVIDLELOC):** This register contains the location pointer to the DevIdle register in MMIO space, as an offset from the specified BAR. The value of this register is not valid, if the Valid bit is not set.
3:1	0h RO	**Base Address Register Number (BARNUM):** Contains the 0's based AR Number of the BAR which contains the location of the SW LTR MMIO register. When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number. Bar 0 is the 1st BAR. The value of this register is not valid, if the Valid bit is not set.
0	1h RO	**VALID:** Set to 1'b1 to indicate that the function has implemented a SW LTR register as specified DevIdle that can be located using the DEVIDLELOC register and BARNUM. Set to 1'b0 to indicate that the function has not implemented a DevIdle register that is compliant to the DevIdle definition. This could be because the function has not implemented SW LTR at all, or has a device specific version of device idle.

21.39 Device Idle Power On Latency (HECI1_DEVIDLEPOL) — Offset B4h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + B4h	D38 h

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	**Reserved (RSVD_15_13):** Reserved
12:10	3h RW/O	**Power On Latency Scale (POLS):** Latency Scale multiplier: 010: 1us 011: 32us All other settings are reserved (not enforced by HW). The value of this register is multiplied with the Power On Latency Value (POLV) to provide the DevIdle power on exit latency multiplier scale from 1us to 32ms. This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value. This register is undefined if the device idle valid bit is 1'b0.
9:0	138h RW/O	**Power On Latency Value (POLV):** 10-bit value that is multiplied by the Power On Latency Scale. A value of 0 indicates a power on latency of less than 1us. This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value. This register is undefined if the device idle valid bit is 1'b0.

21.40 DevIdle Power Control Enabled Register (HECI1_PWRCTRLEN) — Offset B6h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + B6h	E h

Bit Range	Default & Access	Field Name (ID): Description
15:6	0h RO	**Reserved (RSVD_15_6):** Reserved
5	0h RO	**Hardware Autonomous Enable (HAE):** CSE does not support autonomous power gating on idle.
4	0h RO	**Reserved (RSVD_4_4):** Reserved
3	1h RO	**Sleep Enable (SE):** CSE always asserts the sleep signal during power gating.
2	1h RO	**D3-Hot Enable (D3HEN):** CSE could power gate when idle and the PMCSR[1:0] register in the function =2'b11 (D3).
1	1h RO	**DEVIDLE Enable (DEVIDLEN):** CSE could power gate when idle and the DevIdle register (DevIdleC[2] = 1'b1) is set.
0	0h RO	**PMC Request Enable (PMCRE):** CSE will not power gate due to pmc_ip_sw_pg_req_b = LOW.

21.41 Host Extend Register Status (HECI1_HERS) — Offset BCh

This register is used to communicate the CSE FW measurement status information to host.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + BCh	40000000 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Extend Register Valid (ERV):** Set by FW after all FW has been loaded and the measurement data has been stored in HERx registers. If ERA field is SHA-1, the result of the extend operation is in HER:5-1. If ERA field is SHA-256, the result is in HER:8-1.
30	1h RO	**Extend Feature Present (EFP):** This bit is hardwired to 1 to allow driver software to easily detect the chipset supports the Extend Registers FW measurement feature.
29:4	0h RO	**Reserved (RSVD_29_4):** Reserved
3:0	0h RO/V	**Extend Register Algorithm (ERA):** This field indicates the hash algorithm used in the FW measurement Extend operation. Encodings are: 0x0: SHA-1, 0x2: SHA-256. Other values: reserved. This field is set by FW with the used hash algorithm value when the ERV bit is set to 1. This field is meaningless when the ERV bit is 0. This field does NOT have any defined reset value.

21.42 Host Extend Register DW1 (HECI1_HER1) — Offset C0h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + C0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Host Extend Register DW1 (ERDW1):** Host Extend Register DW1.

21.43 Host Extend Register DW2 (HECI1_HER2) — Offset C4h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + C4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Host Extend Register DW2 (ERDW2):** Host Extend Register DW2.

21.44 Host Extend Register DW3 (HECI1_HER3) — Offset C8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + C8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Host Extend Register DW3 (ERDW3):** Host Extend Register DW3.

21.45 Host Extend Register DW4 (HECI1_HER4) — Offset CCh

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + CCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Host Extend Register DW4 (ERDW4):** Host Extend Register DW4.

21.46 Host Extend Register DW5 (HECI1_HER5) — Offset D0h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + D0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Host Extend Register DW5 (ERDW5):** Host Extend Register DW5.

21.47 Host Extend Register DW6 (HECI1_HER6) — Offset D4h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + D4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Host Extend Register DW6 (ERDW6):** Host Extend Register DW6.

21.48 Host Extend Register DW7 (HECI1_HER7) — Offset D8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + D8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Host Extend Register DW7 (ERDW7):** Host Extend Register DW7.

21.49 Host Extend Register DW8 (HECI1_HER8) — Offset DCh

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + DCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Host Extend Register DW8 (ERDW8):** Host Extend Register DW8.

21.50 Manufacturer's ID (HECI1_MANID) — Offset F8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + F8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD_31_28):** Reserved
27:24	0h RO/V	**Dot Portion of Process ID (DPOP):** Indicates the dot process. This field is set by a special IOSF SB message (SetIDValue).
23:16	0h RO/V	**Manufacturing Stepping ID (MSID):** This field is incremented for each stepping of the part. Note that this field can be used by software to differentiate steppings when Revision ID may not change. This field is set by a special IOSF SB message (SetIDValue).
15:8	0h RO/V	**Manufacturing ID (MID):** This field is set by a special IOSF SB message (SetIDValue).
7:0	0h RO/V	**Process Portion of Process ID (PPOP):** Indicates the process. The dot portion of the process is reflected in bits [27:24]. Note that this field can be used by software to differentiate steppings when Revision ID may not change. This field is set by a special IOSF SB message (SetIDValue).

§ §

22 HECI1 Registers

This chapter documents the registers in Bus: 0, Device 15, Function 0.

22.1 Identifiers (HECI1_ID) — Offset 0h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 0h	8086 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO/V	**Device ID (DID):** Indicates device number assigned by Intel. The upper 9 bits of this field are set by a special sideband message (SetIDValue). The lower 7 bits are set by soft straps. Note that the default value of this field is not 0, but depends on the value of soft straps. Refer to CSE spec in order to find the value of 7 LSBs.
15:0	8086h RO	**Vendor ID (VID):** 16-bit field which indicates Intel is the vendor, assigned by the PCI SIG.

22.2 Command (HECI1_CMD) — Offset 4h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + 4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:11	0h RO	**Reserved (RSVD):** Reserved
10	0h RW	**Interrupt Disable (ID):** Disables this device from generating PCI line based interrupts. This bit does not have any effect on MSI operation.
9	0h RO	**Fast Back-to-Back Enable (FBE):** Not implemented, hardwired to 0.
8	0h RO	**SERR# Enable (SEE):** Not implemented, hardwired to 0.
7	0h RO	**Wait Cycle Enable (WCC):** Not implemented, hardwired to 0.
6	0h RO	**Parity Error Response Enable (PEE):** Not implemented, hardwired to 0.
5	0h RO	**VGA Palette Snooping Enable (VAG):** Not implemented, hardwired to 0
4	0h RO	**Memory Write And Invalidate Enable (MWIE):** Not implemented, hardwired to 0.
3	0h RO	**Special Cycle Enable (SCE):** Not implemented, hardwired to 0.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Bus Master Enable (BME):** Controls the HECI host controller's abiity to act as a system memory master for data transfers. When this bit is cleared, HECI bus master activity stops and any active DMA engines return to an idle condition. This bit is made visible to firmware through the H_PCI_CSR register, and changes to this bit may be configured by the H_PCI_CSR register to generate an MSI to mIA. When this bit is '0', HECI is blocked from generating MSI to the host CPU. Note that this bit does not block HECI accesses to CSE-UMA.
1	0h RW	**Memory Space Enable (MSE):** Controls access to the HECI host controller's memory mapped register space.
0	0h RO	**I/O Space Enable (IOSE):** Not implemented, hardwired to 0.

22.3 Status (HECI1_STS) — Offset 6h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + 6h	10 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**Detected Parity Error (DPE):** Not implemented, hardwired to 0.
14	0h RO	**Signaled System Error (SSE):** Not implemented, hardwired to 0.
13	0h RO	**Received Master-Abort (RMA):** Not implemented, hardwired to 0.
12	0h RO	**Received Target Abort (RTA):** Not implemented, hardwired to 0.
11	0h RO	**Signaled Target-Abort (STA):** Not implemented, hardwired to 0.
10:9	0h RO	**DEVSEL# Timing (DEVT):** These bits are hardwired to 00.
8	0h RO	**Master Data Pariy Error Detected (DPD):** Not implemented, hardwired to 0.
7	0h RO	**Fast Back-to-Back Capable (FBC):** Not implemented, hardwired to 0.
6	0h RO	**Reserved (RSVD_6_6):** Reserved
5	0h RO	**66 MHz Capable (C66):** Not implemented, hardwired to 0.
4	1h RO	**Capabilities List (CL):** Indicates the presence of a capabilities list, hardwired to 1.
3	0h RO/V	**Interrupt Status (IS):** Reflects the state of the INTx# signal at the input of the enable/disable circuit. This bit is a 1 when the INTx# is asserted. This bit is a 0 after the interrupt is cleared (independent of the state of the Interrupt Disable bit in the command register). Note that this bit is not set by an MSI.
2:0	0h RO	**Reserved (RSVD_2_0):** Reserved

22.4 Revision ID And Class Code (HECI1_RID_CC) — Offset 8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 8h	7800000 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	7h RO	**Base Class Code (BCC):** Indicates the base class code of the HECI host controller device.
23:16	80h RO	**Sub Class Code (SCC):** Indicates the sub class code of the HECI host controller device.
15:8	0h RO	**Programming Interface (PI):** Indicates the programming interface of the HECI host controller device.
7:0	0h RO/V	**Revision ID (RID):** Indicates stepping of the HECI host controller. This field is set by a special IOSF SB message (SetIDValue).

22.5 Cache Line Size (HECI1_CLS) — Offset Ch

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:0] + Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Cache Line Size (CLS):** Not implemented, hardwired to 0.

22.6 Master Latency Timer (HECI1_MLT) — Offset Dh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:0] + Dh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Master Latency Timer (MLT):** Not implemented, hardwired to 0.

22.7 Header Type (HECI1_HTYPE) — Offset Eh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:0] + Eh	80 h

Bit Range	Default & Access	Field Name (ID): Description
7	1h RO	**Multi-Function Device (MFD):** Indicates the HECI host controller is part of a multi-function device.
6:0	0h RO	**Header Layout (HL):** Indicates that the HECI host controller uses a target device layout.

22.8 Built In Self-Test (HECI1_BIST) — Offset Fh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:0] + Fh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**BIST Capable (BC):** Not implemented, hardwired to 0.
6:0	0h RO	**Reserved (RSVD):** Reserved

22.9 HECI MMIO Base Address Low (HECI1_MMIO_MBAR_LO) — Offset 10h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 10h	4 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**Base Address Low (BA_LO):** Lower 32 bits of base address of register memory space.
11:4	0h RO	**Memory Size (MS):** This field is hardwired to 0 to indicate that HECI device requests 4KB of memory space.
3	0h RO	**Prefetchable (PF):** Indicates that this range is not pre-fetchable.
2:1	2h RO	**Type (TP):** Indicates that this range can be mapped anywhere in 64-bit address space.
0	0h RO	**Resource Type Indicator (RTE):** Indicates a request for register memory space.

22.10 HECI MMIO Base Address High (HECI1_MMIO_MBAR_HI) — Offset 14h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Base Address High (BA_HI):** Upper 32 bits of base address of register memory space.

22.11 BAR1 — Offset 18h

BAR1 -Base Address Register1. Not in use in CSE

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 18h	0 h

Bit Range	Default & Access	Field Name (ID): Description
Reserved		

22.12 BAR1_HIGH — Offset 1Ch

BAR1 -Base Address Register1 High. Not in use in CSE

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 1Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
Reserved		

22.13 Sub System Identifiers (HECI1_SS) — Offset 2Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**Subsystem ID (SSID):** Indicates the subsystem identifier. This field should be programmed by BIOS during boot-up. Once written, this register becomes Read Only.
15:0	0h RW/O	**Subsystem Vendor ID (SSVID):** Indicates the subsystem vendor identifier. This field should be programmed by BIOS during boot- up. Once written, this register becomes Read Only.

22.14 EXPANSION_ROM_BASEADDR — Offset 30h

EXPANSION ROM base address

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 30h	0 h

Bit Range	Default & Access	Field Name (ID): Description
Reserved		

22.15 Capabilities Pointer (HECI1_CAP) — Offset 34h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 34h	50 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD):** Reserved
7:0	50h RO	**Capability Pointer (CP):** Indicates the first capability pointer offset. It points to the PCI power management capability offset.

22.16 Interrupt Information (HECI1_INTR) — Offset 3Ch

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + 3Ch	100 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	1h RO/V	**Interrupt Pin (IPIN):** This field indicates the virtual interrupt pin the HECI host controller uses. A value of 0x1/0x2/0x3/0x4 indicates that this function implements legacy interrupt on INTA/INTB/INTC/INTD, respectively. When legacy interrupt is not used this register should be set to 0x0. Programmed by BIOS over IOSF SB through corresponding Private CR register.

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Interrupt Line (ILINE):** Software written value to indicate which interrupt line (vector) the interrupt is connected to. No hardware action is taken on this field.

22.17 Minimum Grant (HECI1_MGNT) — Offset 3Eh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:0] + 3Eh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Grant (GNT):** Not implemented, hardwired to 0.

22.18 Maximum Latency (HECI1_MLAT) — Offset 3Fh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:0] + 3Fh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**LAT:** Not implemented, hardwired to 0.

22.19 Host Firmware Status (HECI1_HFS) — Offset 40h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 40h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Firmware Status Host Access (FS_HA):** Indicates current status of the firmware for the HECI controller. This field is the host's read only access to the FS field in the CSE Firmware Status register.

22.20 Miscellaneous Shadow (HECI1_MISC_SHDW) — Offset 44h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 44h	80000000 h

Bit Range	Default & Access	Field Name (ID): Description
31	1h RO	**Miscellaneous Shadow Valid (MSVLD):** This bit is hardwired to 1 to indicate that this HECI device implements the Miscellaneous Shadow register. This bit can be used by host software that is bus/dev/function number agnostic (such as HECI operating system driver) to discover whether the Miscellaneous Shadow register is implemented or not.
30:17	0h RO	**Reserved (RSVD_30_17):** Reserved
16	0h RO/V	**CSE UMA Size Valid (CUSZV):** This bit indicates that FW has written the CUSZ field. This field reflects the value of CSE CUBA.CUSZV.
15:7	0h RO	**Reserved (RSVD_15_7):** Reserved
6:0	0h RO/V	**CSE UMA Size (CUSZ):** These bits reflect firmware's desired size of CSEUMA memory region. It is configured by firmware prior to bring up core power and allowing BIOS to initialize memory. The legal CSEUMA sizes and encodings for this field are: 0000000b (0MB, no CSEUMA region) 0000001b (1MB) 0000010b (2MB) 0000100b (4MB) 0001000b (8MB) 0010000b (16MB) 0100000b (32MB) 1000000b (64MB). This field reflects the value of CSE CUBA.CUSZ.

22.21 General Status Shadow 1 (HECI1_GS_SHDW1) — Offset 48h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 48h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 1 (GSS1):** This field is host side shadow of CSE General Status 1 (CSE_GS1).

22.22 Host General Status (HECI1_H_GS1) — Offset 4Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 4Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Host General Status (H_GS1):** General status of Host. This field is not used by hardware.

22.23 PCI Power Management Capability ID (HECI1_PID) — Offset 50h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + 50h	8C01 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	8Ch RO	**NEXT:** Indicates the location of the next capability item in the list. This is the Message Signaled Interrupts capability.
7:0	1h RO	**Cap ID (CID):** Indicates that this pointer is a PCI power management.

22.24 PCI Power Management Capabilities (HECI1_PC) — Offset 52h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + 52h	4003 h

Bit Range	Default & Access	Field Name (ID): Description
15:11	8h RO	**PME Support (PSUP):** Indicates the states that can generate PME#. HECI can assert PME# from D3hot only.
10	0h RO	**D2 Support (D2S):** The D2 state is not supported for the HECI host controller.
9	0h RO	**D1 Support (D1S):** The D1 state is not supported for the HECI host controller.
8:6	0h RO	**Aux Current (AUXC):** Reports the maximum Suspend well current required when in the D3COLD state. Value of 0 is reported.
5	0h RO	**Device Specific Initialization (DSI):** Indicates whether device-specific initialization is required.
4	0h RO	**Reserved (RSVD):** Reserved
3	0h RO	**PME Clock (PMEC):** Indicates that PCI clock is not required to generate PME#.
2:0	3h RO	**Version (VS):** Indicates support for Revision 1.2 of the PCI Power Management Specification.

22.25 PCI Power Management Control And Status (HECI1_PMCS) — Offset 54h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + 54h	8 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C/V	**PME Status (PMES):** The PME Status bit in HECI space can be set to '1' by CSE FW performing a write into CSE side register to set PMES. This bit is cleared by host CPU writing a '1' to it. CSE cannot clear this bit. Host CPU writes with value '0' have no effect on this bit.
14:9	0h RO	**Reserved (RSVD_14_9):** Reserved
8	0h RW	**PME Enable (PMEE):** When set, PME_assert and PME_deassert messages are sent over Sideband to PMC based on the PMES bit. When reset these messages may not be sent.
7:4	0h RO	**Reserved (RSVD_7_4):** Reserved
3	1h RO	**No Soft Reset (NSR):** This bit indicates that when the HECI host controller is transitioning from D3hot to D0 due to power state command, it does not perform an internal reset.
2	0h RO	**Reserved (RSVD_2_2):** Reserved
1:0	0h RW	**Power State (PS):** This field is used both to determine the current power state of the HECI host controller and to set a new power state. The values are: 00 - D0 state 11 - D3HOT state. The D1 and D2 states are not supported for this HECI host controller. If software attempts to write an unsupported, optional state to this field, the write operation must complete normally on the bus; however, the data is discarded and no state change occurs. When in the D3HOT state, the HBA's configuration space is available, but the registers memory space is not. Additionally, interrupts are blocked. This field is visible to firmware through the H_PCI_CSR register, and changes to this field may be configured by the H_PCI_CSR register to generate an MSI to mIA.

22.26 General Status Shadow 2 (HECI1_GS_SHDW2) — Offset 60h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 60h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 2 (GSS2):** This field is host side shadow of CSE General Status 2 (CSE_GS2).

22.27 General Status Shadow 3 (HECI1_GS_SHDW3) — Offset 64h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 64h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 3 (GSS3):** This field is host side shadow of CSE General Status 3 (CSE_GS3).

22.28 General Status Shadow 4 (HECI1_GS_SHDW4) — Offset 68h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 68h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 4 (GSS4):** This field is host side shadow of CSE General Status 4 (CSE_GS4).

22.29 General Status Shadow 5 (HECI1_GS_SHDW5) — Offset 6Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 6Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 5 (GSS5):** This field is host side shadow of CSE General Status 5 (CSE_GS5).

22.30 Host General Status 2 (HECI1_H_GS2) — Offset 70h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 70h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Host General Status 2 (H_GS2):** General status of Host. This field is not used by hardware.

22.31 Host General Status 3 (HECI1_H_GS3) — Offset 74h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 74h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Host General Status 3 (H_GS3):** General status of Host. This field is not used by hardware.

22.32 POWERCAPID — Offset 80h

POWERCAPID - PowerManagement Capability ID. Not in use in CSE

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 80h	0 h

Bit Range	Default & Access	Field Name (ID): Description
Reserved		

22.33 PMECTRLSTATUS — Offset 84h

Not in use in CSE

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 84h	0 h

Bit Range	Default & Access	Field Name (ID): Description
Reserved		

22.34 Message Signaled Interrupt Identifiers (HECI1_MID) — Offset 8Ch

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + 8Ch	A405 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	A4h RO	**NEXT:** Indicates the location of the next capability item in the list. This is the DevIdle conventional PCI vendor specific capability.
7:0	5h RO	**Capability ID (CID):** Indicates MSI.

22.35 Message Signaled Interrupt Message Control (HECI1_MC) — Offset 8Eh

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + 8Eh	80 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	**Reserved (RSVD):** Reserved
7	1h RO	**64 Bit Address Capable (C64):** Specifies whether capable of generating 64-bit messages.
6:4	0h RO	**Multiple Message Enable (MME):** Not implemented, hardwired to 0.
3:1	0h RO	**Multiple Message Capable (MMC):** Not implemented, hardwired to 0.
0	0h RW	**MSI Enable (MSIE):** If set, MSI is enabled and INTx virtual wires are not used to generate interrupts.

22.36 Message Signaled Interrupt Message Address (HECI1_MA) — Offset 90h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 90h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**ADDR:** Lower 32 bits of the system specified message address, always DW aligned.

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RO	**Reserved (RSVD):** Reserved

22.37 Message Signaled Interrupt Upper Address (HECI1_MUA) — Offset 94h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 94h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Upper Address (UADDR):** Upper 32 bits of the system specified message address. This register is optional and only implemented if MC.C64=1.

22.38 Message Signaled Interrupt Message Data (HECI1_MD) — Offset 98h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + 98h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW	**DATA:** This 16-bit field is programmed by system software if MSI is enabled. Its content is driven onto the FSB during the data phase of the MSI memory write transaction.

22.39 DEVICE_IDLE_POINTER_REG — Offset 9Ch

Device IDLE pointer register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + 9Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
Reserved		

22.40 HECI Interrupt Delivery Mode (HECI1_HIDM) — Offset A0h

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:0] + A0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:3	0h RO	**Reserved (RSVD):** Reserved
2	0h RW/1S/V	**HIDM Lock (HIDM_L):** Writing 1 to this bit locks the HIDM field.
1:0	0h RW/L	**HECI Interrupt Delivery Mode (HIDM):** These bits control what type of interrupt the HECI will send when CSE FW writes to set the CSE_IG bit. They are interpreted as follows: 00 - Generate Legacy or MSI interrupt; 01 - Generate SCI; 10 - Generate SMI; This field may be locked by writing 1 to HIDM_L bit.

22.41 Vendor Specific Capability Register (HECI1_VSCR) — Offset A4h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + A4h	F0140009 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RO	**Vendor-Specific Capability ID (VSID):** A value of 4'hF in this 4-bit field identifies this Vendor Specific Capability as an Extended Capability, which uses a VSEC 16-bit Extended Capability ID in the subsequent four bytes. This field allows software to differentiate this capability from other Vendor Specific capabilities.
27:24	0h RO	**Vendor Specific Capability Revision (VSREV):** For a VSID of 4'hF, this field is reserved 4'h0.
23:16	14h RO	**Vendor Specific Capability Length (VSLEN):** This field indicates the number of bytes of this Vendor Specific capability as required by the PCI spec inclusive of the CapID and Next Pointer. It has the value of 8'h14 for the DevIdle Capability.
15:8	0h RO	**NEXT:** Pointer to the configuration offset of the next Capability item. It is 8'h00 since it is the final item in the Capability list.
7:0	9h RO	**Capability ID (CAPID):** The value of 8h09 in this field indicates a Vendor Specific capability.

22.42 Vendor Specific Extended Capability Register (HECI1_VSEC) — Offset A8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + A8h	1400010 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	14h RO	**Vendor Specific Extended Capability Length (VSECLEN):** This field indicates the number of bytes of this Vendor Specific capability inclusive of the 1st DW which includes the capability ID and Next pointer. For DevIdle, this is 12'h14.
19:16	0h RO	**Vendor Specific Extended Capability Revision (VSECREV):** For this revision of DevIdle, this field is 4h0.
15:0	10h RO	**Vendor Specific Extended Capability ID (NEXT):** DevIdle has been assigned the Intel VSEC ID of 16h10.

22.43 SW LTR Pointer Register (HECI1_SWLTRPTR) — Offset ACh

Software Latency Tolerance Reporting (LTR) Pointer Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + ACh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**SW LTR Update MMIO Offset Location (SWLTRLOC):** This register contains the location pointer to the SW LTR register in MMIO space, as an offset from the specified BAR. The value of this register is not valid, if the Valid bit is not set.
3:1	0h RO	**Base Address Register Number (BARNUM):** Contains the 0's based AR Number of the BAR which contains the location of the SW LTR MMIO register. When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number. Bar 0 is the 1st BAR. The value of this register is not valid, if the Valid bit is not set.
0	0h RO	**VALID:** Set to 1'b1 to indicate that the function has implemented a SW LTR register as specified in the DevIdle that can be located using the SWLTRLOC register and BARNUM. Set to 1'b0 to indicate that the function has not implemented a SW LTR register that is compliant to the DevIdle definition. This could be because the function has not implemented SW LTR at all, or has a device specific version of SW LTR.

22.44 Device Idle Pointer Register (HECI1_DEVIDLEPTR) — Offset B0h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + B0h	8001 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	800h RO	**Device Idle MMIO Offset Location (DEVIDLELOC):** This register contains the location pointer to the DevIdle register in MMIO space, as an offset from the specified BAR. The value of this register is not valid, if the Valid bit is not set.
3:1	0h RO	**Base Address Register Number (BARNUM):** Contains the 0's based AR Number of the BAR which contains the location of the SW LTR MMIO register. When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number. Bar 0 is the 1st BAR. The value of this register is not valid, if the Valid bit is not set.

Bit Range	Default & Access	Field Name (ID): Description
0	1h RO	**VALID:** Set to 1'b1 to indicate that the function has implemented a SW LTR register as specified DevIdle that can be located using the DEVIDLELOC register and BARNUM. Set to 1'b0 to indicate that the function has not implemented a DevIdle register that is compliant to the DevIdle definition. This could be because the function has not implemented SW LTR at all, or has a device specific version of device idle.

22.45 Device Idle Power On Latency (HECI1_DEVIDLEPOL) — Offset B4h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + B4h	D38 h

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	**Reserved (RSVD_15_13):** Reserved
12:10	3h RW/O	**Power On Latency Scale (POLS):** Latency Scale multiplier: 010: 1us 011: 32us All other settings are reserved (not enforced by HW). The value of this register is multiplied with the Power On Latency Value (POLV) to provide the DevIdle power on exit latency multiplier scale from 1us to 32ms. This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value. This register is undefined if the device idle valid bit is 1'b0.
9:0	138h RW/O	**Power On Latency Value (POLV):** 10-bit value that is multiplied by the Power On Latency Scale. A value of 0 indicates a power on latency of less than 1us. This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value. This register is undefined if the device idle valid bit is 1'b0.

22.46 DevIdle Power Control Enabled Register (HECI1_PWRCTRLEN) — Offset B6h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:0] + B6h	E h

Bit Range	Default & Access	Field Name (ID): Description
15:6	0h RO	**Reserved (RSVD_15_6):** Reserved
5	0h RO	**Hardware Autonomous Enable (HAE):** CSE does not support autonomous power gating on idle.
4	0h RO	**Reserved (RSVD_4_4):** Reserved
3	1h RO	**Sleep Enable (SE):** CSE always asserts the sleep signal during power gating.
2	1h RO	**D3-Hot Enable (D3HEN):** CSE could power gate when idle and the PMCSR[1:0] register in the function =2'b11 (D3).

Bit Range	Default & Access	Field Name (ID): Description
1	1h RO	**DEVIDLE Enable (DEVIDLEN):** CSE could power gate when idle and the DevIdle register (DevIdleC[2] = 1'b1) is set.
0	0h RO	**PMC Request Enable (PMCRE):** CSE will not power gate due to pmc_ip_sw_pg_req_b = LOW.

22.47 GEN_REGRW3 — Offset B8h

General Purpose Read Write Register3. Not in use in CSE

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + B8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
Reserved		

22.48 Host Extend Register Status (HECI1_HERS) — Offset BCh

This register is used to communicate the CSE FW measurement status information to host.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + BCh	40000000 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Extend Register Valid (ERV):** Set by FW after all FW has been loaded and the measurement data has been stored in HERx registers. If ERA field is SHA-1, the result of the extend operation is in HER:5-1. If ERA field is SHA-256, the result is in HER:8-1.
30	1h RO	**Extend Feature Present (EFP):** This bit is hardwired to 1 to allow driver software to easily detect the chipset supports the Extend Registers FW measurement feature.
29:4	0h RO	**Reserved (RSVD_29_4):** Reserved
3:0	0h RO/V	**Extend Register Algorithm (ERA):** This field indicates the hash algorithm used in the FW measurement Extend operation. Encodings are: 0x0: SHA-1, 0x2: SHA-256. Other values: reserved. This field is set by FW with the used hash algorithm value when the ERV bit is set to 1. This field is meaningless when the ERV bit is 0. This field does NOT have any defined reset value.

22.49 Host Extend Register DW1 (HECI1_HER1) — Offset C0h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + C0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Host Extend Register DW1 (ERDW1):** Host Extend Register DW1.

22.50 Host Extend Register DW2 (HECI1_HER2) — Offset C4h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + C4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Host Extend Register DW2 (ERDW2):** Host Extend Register DW2.

22.51 Host Extend Register DW3 (HECI1_HER3) — Offset C8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + C8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Host Extend Register DW3 (ERDW3):** Host Extend Register DW3.

22.52 Host Extend Register DW4 (HECI1_HER4) — Offset CCh

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + CCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Host Extend Register DW4 (ERDW4):** Host Extend Register DW4.

22.53 Host Extend Register DW5 (HECI1_HER5) — Offset D0h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + D0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Host Extend Register DW5 (ERDW5):** Host Extend Register DW5.

22.54 Host Extend Register DW6 (HECI1_HER6) — Offset D4h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + D4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Host Extend Register DW6 (ERDW6):** Host Extend Register DW6.

22.55 Host Extend Register DW7 (HECI1_HER7) — Offset D8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + D8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Host Extend Register DW7 (ERDW7):** Host Extend Register DW7.

22.56 Host Extend Register DW8 (HECI1_HER8) — Offset DCh

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + DCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Host Extend Register DW8 (ERDW8):** Host Extend Register DW8.

22.57 Manufacturer's ID (HECI1_MANID) — Offset F8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:0] + F8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD_31_28):** Reserved
27:24	0h RO/V	**Dot Portion of Process ID (DPOP):** Indicates the dot process. This field is set by a special IOSF SB message (SetIDValue).
23:16	0h RO/V	**Manufacturing Stepping ID (MSID):** This field is incremented for each stepping of the part. Note that this field can be used by software to differentiate steppings when Revision ID may not change. This field is set by a special IOSF SB message (SetIDValue).
15:8	0h RO/V	**Manufacturing ID (MID):** This field is set by a special IOSF SB message (SetIDValue).
7:0	0h RO/V	**Process Portion of Process ID (PPOP):** Indicates the process. The dot portion of the process is reflected in bits [27:24]. Note that this field can be used by software to differentiate steppings when Revision ID may not change. This field is set by a special IOSF SB message (SetIDValue).

§ §

23 HECI2 Registers

This chapter documents the registers in Bus: 0, Device 15, Function 1.

23.1 Identifiers (HECI2_ID) — Offset 0h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + 0h	8086 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO/V	**Device ID (DID):** Indicates device number assigned by Intel. The upper 9 bits of this field are set by a special sideband message (SetIDValue). The lower 7 bits are set by soft straps. Note that the default value of this field is not 0, but depends on the value of soft straps. Refer to CSE spec in order to find the value of 7 LSBs.
15:0	8086h RO	**Vendor ID (VID):** 16-bit field which indicates Intel is the vendor, assigned by the PCI SIG.

23.2 Command (HECI2_CMD) — Offset 4h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:1] + 4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:11	0h RO	**Reserved (RSVD):** Reserved
10	0h RW	**Interrupt Disable (ID):** Disables this device from generating PCI line based interrupts. This bit does not have any effect on MSI operation.
9	0h RO	**Fast Back-to-Back Enable (FBE):** Not implemented, hardwired to 0.
8	0h RO	**SERR# Enable (SEE):** Not implemented, hardwired to 0.
7	0h RO	**Wait Cycle Enable (WCC):** Not implemented, hardwired to 0.
6	0h RO	**Parity Error Response Enable (PEE):** Not implemented, hardwired to 0.
5	0h RO	**VGA Palette Snooping Enable (VAG):** Not implemented, hardwired to 0
4	0h RO	**Memory Write And Invalidate Enable (MWIE):** Not implemented, hardwired to 0.
3	0h RO	**Special Cycle Enable (SCE):** Not implemented, hardwired to 0.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Bus Master Enable (BME):** Controls the HECI host controller's abiity to act as a system memory master for data transfers. When this bit is cleared, HECI bus master activity stops and any active DMA engines return to an idle condition. This bit is made visible to firmware through the H_PCI_CSR register, and changes to this bit may be configured by the H_PCI_CSR register to generate an MSI to mIA. When this bit is '0', HECI is blocked from generating MSI to the host CPU. Note that this bit does not block HECI accesses to CSE-UMA.
1	0h RW	**Memory Space Enable (MSE):** Controls access to the HECI host controller's memory mapped register space.
0	0h RO	**I/O Space Enable (IOSE):** Not implemented, hardwired to 0.

23.3 Status (HECI2_STS) — Offset 6h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:1] + 6h	10 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**Detected Parity Error (DPE):** Not implemented, hardwired to 0.
14	0h RO	**Signaled System Error (SSE):** Not implemented, hardwired to 0.
13	0h RO	**Received Master-Abort (RMA):** Not implemented, hardwired to 0.
12	0h RO	**Received Target Abort (RTA):** Not implemented, hardwired to 0.
11	0h RO	**Signaled Target-Abort (STA):** Not implemented, hardwired to 0.
10:9	0h RO	**DEVSEL# Timing (DEVT):** These bits are hardwired to 00.
8	0h RO	**Master Data Pariy Error Detected (DPD):** Not implemented, hardwired to 0.
7	0h RO	**Fast Back-to-Back Capable (FBC):** Not implemented, hardwired to 0.
6	0h RO	**Reserved (RSVD_6_6):** Reserved
5	0h RO	**66 MHz Capable (C66):** Not implemented, hardwired to 0.
4	1h RO	**Capabilities List (CL):** Indicates the presence of a capabilities list, hardwired to 1.
3	0h RO/V	**Interrupt Status (IS):** Reflects the state of the INTx# signal at the input of the enable/disable circuit. This bit is a 1 when the INTx# is asserted. This bit is a 0 after the interrupt is cleared (independent of the state of the Interrupt Disable bit in the command register). Note that this bit is not set by an MSI.
2:0	0h RO	**Reserved (RSVD_2_0):** Reserved

23.4 Revision ID And Class Code (HECI2_RID_CC) — Offset 8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + 8h	7800000 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	7h RO	**Base Class Code (BCC):** Indicates the base class code of the HECI host controller device.
23:16	80h RO	**Sub Class Code (SCC):** Indicates the sub class code of the HECI host controller device.
15:8	0h RO	**Programming Interface (PI):** Indicates the programming interface of the HECI host controller device.
7:0	0h RO/V	**Revision ID (RID):** Indicates stepping of the HECI host controller. This field is set by a special IOSF SB message (SetIDValue).

23.5 Cache Line Size (HECI2_CLS) — Offset Ch

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:1] + Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Cache Line Size (CLS):** Not implemented, hardwired to 0.

23.6 Master Latency Timer (HECI2_MLT) — Offset Dh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:1] + Dh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Master Latency Timer (MLT):** Not implemented, hardwired to 0.

23.7 Header Type (HECI2_HTYPE) — Offset Eh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:1] + Eh	80 h

Bit Range	Default & Access	Field Name (ID): Description
7	1h RO	**Multi-Function Device (MFD):** Indicates the HECI host controller is part of a multi-function device.
6:0	0h RO	**Header Layout (HL):** Indicates that the HECI host controller uses a target device layout.

23.8 Built In Self-Test (HECI2_BIST) — Offset Fh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:1] + Fh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**BIST Capable (BC):** Not implemented, hardwired to 0.
6:0	0h RO	**Reserved (RSVD):** Reserved

23.9 HECI MMIO Base Address Low (HECI2_MMIO_MBAR_LO) — Offset 10h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + 10h	4 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**Base Address Low (BA_LO):** Lower 32 bits of base address of register memory space.
11:4	0h RO	**Memory Size (MS):** This field is hardwired to 0 to indicate that HECI device requests 4KB of memory space.
3	0h RO	**Prefetchable (PF):** Indicates that this range is not pre-fetchable.
2:1	2h RO	**Type (TP):** Indicates that this range can be mapped anywhere in 64-bit address space.
0	0h RO	**Resource Type Indicator (RTE):** Indicates a request for register memory space.

23.10 HECI MMIO Base Address High (HECI2_MMIO_MBAR_HI) — Offset 14h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + 14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Base Address High (BA_HI):** Upper 32 bits of base address of register memory space.

23.11 Sub System Identifiers (HECI2_SS) — Offset 2Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**Subsystem ID (SSID):** Indicates the subsystem identifier. This field should be programmed by BIOS during boot-up. Once written, this register becomes Read Only.
15:0	0h RW/O	**Subsystem Vendor ID (SSVID):** Indicates the subsystem vendor identifier. This field should be programmed by BIOS during boot- up. Once written, this register becomes Read Only.

23.12 Capabilities Pointer (HECI2_CAP) — Offset 34h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + 34h	50 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD):** Reserved
7:0	50h RO	**Capability Pointer (CP):** Indicates the first capability pointer offset. It points to the PCI power management capability offset.

23.13 Interrupt Information (HECI2_INTR) — Offset 3Ch

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:1] + 3Ch	100 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	1h RO/V	**Interrupt Pin (IPIN):** This field indicates the virtual interrupt pin the HECI host controller uses. A value of 0x1/0x2/0x3/0x4 indicates that this function implements legacy interrupt on INTA/INTB/INTC/INTD, respectively. When legacy interrupt is not used this register should be set to 0x0. Programmed by BIOS over IOSF SB through corresponding Private CR register.
7:0	0h RW	**Interrupt Line (ILINE):** Software written value to indicate which interrupt line (vector) the interrupt is connected to. No hardware action is taken on this field.

23.14 Minimum Grant (HECI2_MGNT) — Offset 3Eh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:1] + 3Eh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Grant (GNT):** Not implemented, hardwired to 0.

23.15 Maximum Latency (HECI2_MLAT) — Offset 3Fh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:1] + 3Fh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**LAT:** Not implemented, hardwired to 0.

23.16 Host Firmware Status (HECI2_HFS) — Offset 40h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + 40h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Firmware Status Host Access (FS_HA):** Indicates current status of the firmware for the HECI controller. This field is the host's read only access to the FS field in the CSE Firmware Status register.

23.17 General Status Shadow 1 (HECI2_GS_SHDW1) — Offset 48h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + 48h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 1 (GSS1):** This field is host side shadow of CSE General Status 1 (CSE_GS1).

23.18 Host General Status (HECI2_H_GS1) — Offset 4Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + 4Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Host General Status (H_GS1):** General status of Host. This field is not used by hardware.

23.19 PCI Power Management Capability ID (HECI2_PID) — Offset 50h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:1] + 50h	8C01 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	8Ch RO	**NEXT:** Indicates the location of the next capability item in the list. This is the Message Signaled Interrupts capability.
7:0	1h RO	**Cap ID (CID):** Indicates that this pointer is a PCI power management.

23.20 PCI Power Management Capabilities (HECI2_PC) — Offset 52h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:1] + 52h	4003 h

Bit Range	Default & Access	Field Name (ID): Description
15:11	8h RO	**PME Support (PSUP):** Indicates the states that can generate PME#. HECI can assert PME# from D3hot only.
10	0h RO	**D2 Support (D2S):** The D2 state is not supported for the HECI host controller.
9	0h RO	**D1 Support (D1S):** The D1 state is not supported for the HECI host controller.
8:6	0h RO	**Aux Current (AUXC):** Reports the maximum Suspend well current required when in the D3COLD state. Value of 0 is reported.
5	0h RO	**Device Specific Initialization (DSI):** Indicates whether device-specific initialization is required.
4	0h RO	**Reserved (RSVD):** Reserved
3	0h RO	**PME Clock (PMEC):** Indicates that PCI clock is not required to generate PME#.
2:0	3h RO	**Version (VS):** Indicates support for Revision 1.2 of the PCI Power Management Specification.

23.21 PCI Power Management Control And Status (HECI2_PMCS) — Offset 54h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:1] + 54h	8 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C/V	**PME Status (PMES):** The PME Status bit in HECI space can be set to '1' by CSE FW performing a write into CSE side register to set PMES. This bit is cleared by host CPU writing a '1' to it. CSE cannot clear this bit. Host CPU writes with value '0' have no effect on this bit.
14:9	0h RO	**Reserved (RSVD_14_9):** Reserved
8	0h RW	**PME Enable (PMEE):** When set, PME_assert and PME_deassert messages are sent over Sideband to PMC based on the PMES bit. When reset these messages may not be sent.
7:4	0h RO	**Reserved (RSVD_7_4):** Reserved
3	1h RO	**No Soft Reset (NSR):** This bit indicates that when the HECI host controller is transitioning from D3hot to D0 due to power state command, it does not perform an internal reset.
2	0h RO	**Reserved (RSVD_2_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Power State (PS):** This field is used both to determine the current power state of the HECI host controller and to set a new power state. The values are: 00 - D0 state 11 - D3HOT state. The D1 and D2 states are not supported for this HECI host controller. If software attempts to write an unsupported, optional state to this field, the write operation must complete normally on the bus; however, the data is discarded and no state change occurs. When in the D3HOT state, the HBA's configuration space is available, but the registers memory space is not. Additionally, interrupts are blocked. This field is visible to firmware through the H_PCI_CSR register, and changes to this field may be configured by the H_PCI_CSR register to generate an MSI to mIA.

23.22 General Status Shadow 2 (HECI2_GS_SHDW2) — Offset 60h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + 60h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 2 (GSS2):** This field is host side shadow of CSE General Status 2 (CSE_GS2).

23.23 General Status Shadow 3 (HECI2_GS_SHDW3) — Offset 64h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + 64h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 3 (GSS3):** This field is host side shadow of CSE General Status 3 (CSE_GS3).

23.24 General Status Shadow 4 (HECI2_GS_SHDW4) — Offset 68h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + 68h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 4 (GSS4):** This field is host side shadow of CSE General Status 4 (CSE_GS4).

23.25 General Status Shadow 5 (HECI2_GS_SHDW5) — Offset 6Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + 6Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 5 (GSS5):** This field is host side shadow of CSE General Status 5 (CSE_GS5).

23.26 Host General Status 2 (HECI2_H_GS2) — Offset 70h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + 70h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Host General Status 2 (H_GS2):** General status of Host. This field is not used by hardware.

23.27 Host General Status 3 (HECI2_H_GS3) — Offset 74h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + 74h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Host General Status 3 (H_GS3):** General status of Host. This field is not used by hardware.

23.28 Message Signaled Interrupt Identifiers (HECI2_MID) — Offset 8Ch

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:1] + 8Ch	A405 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	A4h RO	**NEXT:** Indicates the location of the next capability item in the list. This is the DevIdle conventional PCI vendor specific capability.
7:0	5h RO	**Capability ID (CID):** Indicates MSI.

23.29 Message Signaled Interrupt Message Control (HECI2_MC) — Offset 8Eh

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:1] + 8Eh	80 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	**Reserved (RSVD):** Reserved
7	1h RO	**64 Bit Address Capable (C64):** Specifies whether capable of generating 64-bit messages.
6:4	0h RO	**Multiple Message Enable (MME):** Not implemented, hardwired to 0.
3:1	0h RO	**Multiple Message Capable (MMC):** Not implemented, hardwired to 0.
0	0h RW	**MSI Enable (MSIE):** If set, MSI is enabled and INTx virtual wires are not used to generate interrupts.

23.30 Message Signaled Interrupt Message Address (HECI2_MA) — Offset 90h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + 90h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**ADDR:** Lower 32 bits of the system specified message address, always DW aligned.
1:0	0h RO	**Reserved (RSVD):** Reserved

23.31 Message Signaled Interrupt Upper Address (HECI2_MUA) — Offset 94h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + 94h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Upper Address (UADDR):** Upper 32 bits of the system specified message address. This register is optional and only implemented if MC.C64=1.

23.32 Message Signaled Interrupt Message Data (HECI2_MD) — Offset 98h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:1] + 98h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW	**DATA:** This 16-bit field is programmed by system software if MSI is enabled. Its content is driven onto the FSB during the data phase of the MSI memory write transaction.

23.33 HECI Interrupt Delivery Mode (HECI2_HIDM) — Offset A0h

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:1] + A0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:3	0h RO	**Reserved (RSVD):** Reserved
2	0h RW/1S/V	**HIDM Lock (HIDM_L):** Writing 1 to this bit locks the HIDM field.
1:0	0h RW/L	**HECI Interrupt Delivery Mode (HIDM):** These bits control what type of interrupt the HECI will send when CSE FW writes to set the CSE_IG bit. They are interpreted as follows: 00 - Generate Legacy or MSI interrupt; 01 - Generate SCI; 10 - Generate SMI; This field may be locked by writing 1 to HIDM_L bit.

23.34 Vendor Specific Capability Register (HECI2_VSCR) — Offset A4h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + A4h	F0140009 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RO	**Vendor-Specific Capability ID (VSID):** A value of 4'hF in this 4-bit field identifies this Vendor Specific Capability as an Extended Capability, which uses a VSEC 16-bit Extended Capability ID in the subsequent four bytes. This field allows software to differentiate this capability from other Vendor Specific capabilities.
27:24	0h RO	**Vendor Specific Capability Revision (VSREV):** For a VSID of 4'hF, this field is reserved 4'h0.
23:16	14h RO	**Vendor Specific Capability Length (VSLEN):** This field indicates the number of bytes of this Vendor Specific capability as required by the PCI spec inclusive of the CapID and Next Pointer. It has the value of 8'h14 for the DevIdle Capability.
15:8	0h RO	**NEXT:** Pointer to the configuration offset of the next Capability item. It is 8'h00 since it is the final item in the Capability list.
7:0	9h RO	**Capability ID (CAPID):** The value of 8h09 in this field indicates a Vendor Specific capability.

23.35 Vendor Specific Extended Capability Register (HECI2_VSEC) — Offset A8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + A8h	1400010 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	14h RO	**Vendor Specific Extended Capability Length (VSECLEN):** This field indicates the number of bytes of this Vendor Specific capability inclusive of the 1st DW which includes the capability ID and Next pointer. For DevIdle, this is 12'h14.
19:16	0h RO	**Vendor Specific Extended Capability Revision (VSECREV):** For this revision of DevIdle, this field is 4h0.
15:0	10h RO	**Vendor Specific Extended Capability ID (NEXT):** DevIdle has been assigned the Intel VSEC ID of 16h10.

23.36 SW LTR Pointer Register (HECI2_SWLTRPTR) — Offset ACh

Software Latency Tolerance Reporting (LTR) Pointer Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + ACh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**SW LTR Update MMIO Offset Location (SWLTRLOC):** This register contains the location pointer to the SW LTR register in MMIO space, as an offset from the specified BAR. The value of this register is not valid, if the Valid bit is not set.
3:1	0h RO	**Base Address Register Number (BARNUM):** Contains the 0's based AR Number of the BAR which contains the location of the SW LTR MMIO register. When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number. Bar 0 is the 1st BAR. The value of this register is not valid, if the Valid bit is not set.
0	0h RO	**VALID:** Set to 1'b1 to indicate that the function has implemented a SW LTR register as specified in the DevIdle that can be located using the SWLTRLOC register and BARNUM. Set to 1'b0 to indicate that the function has not implemented a SW LTR register that is compliant to the DevIdle definition. This could be because the function has not implemented SW LTR at all, or has a device specific version of SW LTR.

23.37 Device Idle Pointer Register (HECI2_DEVIDLEPTR) — Offset B0h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + B0h	8001 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	800h RO	**Device Idle MMIO Offset Location (DEVIDLELOC):** This register contains the location pointer to the DevIdle register in MMIO space, as an offset from the specified BAR. The value of this register is not valid, if the Valid bit is not set.
3:1	0h RO	**Base Address Register Number (BARNUM):** Contains the 0's based AR Number of the BAR which contains the location of the SW LTR MMIO register. When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number. Bar 0 is the 1st BAR. The value of this register is not valid, if the Valid bit is not set.
0	1h RO	**VALID:** Set to 1'b1 to indicate that the function has implemented a SW LTR register as specified DevIdle that can be located using the DEVIDLELOC register and BARNUM. Set to 1'b0 to indicate that the function has not implemented a DevIdle register that is compliant to the DevIdle definition. This could be because the function has not implemented SW LTR at all, or has a device specific version of device idle.

23.38 Device Idle Power On Latency (HECI2_DEVIDLEPOL) — Offset B4h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:1] + B4h	D38 h

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	**Reserved (RSVD_15_13):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
12:10	3h RW/O	**Power On Latency Scale (POLS):** Latency Scale multiplier: 010: 1us 011: 32us All other settings are reserved (not enforced by HW). The value of this register is multiplied with the Power On Latency Value (POLV) to provide the DevIdle power on exit latency multiplier scale from 1us to 32ms. This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value. This register is undefined if the device idle valid bit is 1'b0.
9:0	138h RW/O	**Power On Latency Value (POLV):** 10-bit value that is multiplied by the Power On Latency Scale. A value of 0 indicates a power on latency of less than 1us. This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value. This register is undefined if the device idle valid bit is 1'b0.

23.39 DevIdle Power Control Enabled Register (HECI2_PWRCTRLEN) — Offset B6h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:1] + B6h	E h

Bit Range	Default & Access	Field Name (ID): Description
15:6	0h RO	**Reserved (RSVD_15_6):** Reserved
5	0h RO	**Hardware Autonomous Enable (HAE):** CSE does not support autonomous power gating on idle.
4	0h RO	**Reserved (RSVD_4_4):** Reserved
3	1h RO	**Sleep Enable (SE):** CSE always asserts the sleep signal during power gating.
2	1h RO	**D3-Hot Enable (D3HEN):** CSE could power gate when idle and the PMCSR[1:0] register in the function =2'b11 (D3).
1	1h RO	**DEVIDLE Enable (DEVIDLEN):** CSE could power gate when idle and the DevIdle register (DevIdleC[2] = 1'b1) is set.
0	0h RO	**PMC Request Enable (PMCRE):** CSE will not power gate due to pmc_ip_sw_pg_req_b = LOW.

23.40 Manufacturer's ID (HECI2_MANID) — Offset F8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:1] + F8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD_31_28):** Reserved
27:24	0h RO/V	**Dot Portion of Process ID (DPOP):** Indicates the dot process. This field is set by a special IOSF SB message (SetIDValue).

Bit Range	Default & Access	Field Name (ID): Description
23:16	0h RO/V	**Manufacturing Stepping ID (MSID):** This field is incremented for each stepping of the part. Note that this field can be used by software to differentiate steppings when Revision ID may not change. This field is set by a special IOSF SB message (SetIDValue).
15:8	0h RO/V	**Manufacturing ID (MID):** This field is set by a special IOSF SB message (SetIDValue).
7:0	0h RO/V	**Process Portion of Process ID (PPOP):** Indicates the process. The dot portion of the process is reflected in bits [27:24]. Note that this field can be used by software to differentiate steppings when Revision ID may not change. This field is set by a special IOSF SB message (SetIDValue).

§ §

24 HECI3 Registers

This chapter documents the registers in Bus: 0, Device 15, Function 2.

24.1 Identifiers (HECI3_ID) — Offset 0h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + 0h	8086 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO/V	**Device ID (DID):** Indicates device number assigned by Intel. The upper 9 bits of this field are set by a special sideband message (SetIDValue). The lower 7 bits are set by soft straps. Note that the default value of this field is not 0, but depends on the value of soft straps. Refer to CSE spec in order to find the value of 7 LSBs.
15:0	8086h RO	**Vendor ID (VID):** 16-bit field which indicates Intel is the vendor, assigned by the PCI SIG.

24.2 Command (HECI3_CMD) — Offset 4h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:2] + 4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:11	0h RO	**Reserved (RSVD):** Reserved
10	0h RW	**Interrupt Disable (ID):** Disables this device from generating PCI line based interrupts. This bit does not have any effect on MSI operation.
9	0h RO	**Fast Back-to-Back Enable (FBE):** Not implemented, hardwired to 0.
8	0h RO	**SERR# Enable (SEE):** Not implemented, hardwired to 0.
7	0h RO	**Wait Cycle Enable (WCC):** Not implemented, hardwired to 0.
6	0h RO	**Parity Error Response Enable (PEE):** Not implemented, hardwired to 0.
5	0h RO	**VGA Palette Snooping Enable (VAG):** Not implemented, hardwired to 0
4	0h RO	**Memory Write And Invalidate Enable (MWIE):** Not implemented, hardwired to 0.
3	0h RO	**Special Cycle Enable (SCE):** Not implemented, hardwired to 0.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Bus Master Enable (BME):** Controls the HECI host controller's abiity to act as a system memory master for data transfers. When this bit is cleared, HECI bus master activity stops and any active DMA engines return to an idle condition. This bit is made visible to firmware through the H_PCI_CSR register, and changes to this bit may be configured by the H_PCI_CSR register to generate an MSI to mIA. When this bit is '0', HECI is blocked from generating MSI to the host CPU. Note that this bit does not block HECI accesses to CSE-UMA.
1	0h RW	**Memory Space Enable (MSE):** Controls access to the HECI host controller's memory mapped register space.
0	0h RO	**I/O Space Enable (IOSE):** Not implemented, hardwired to 0.

24.3 Status (HECI3_STS) — Offset 6h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:2] + 6h	10 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**Detected Parity Error (DPE):** Not implemented, hardwired to 0.
14	0h RO	**Signaled System Error (SSE):** Not implemented, hardwired to 0.
13	0h RO	**Received Master-Abort (RMA):** Not implemented, hardwired to 0.
12	0h RO	**Received Target Abort (RTA):** Not implemented, hardwired to 0.
11	0h RO	**Signaled Target-Abort (STA):** Not implemented, hardwired to 0.
10:9	0h RO	**DEVSEL# Timing (DEVT):** These bits are hardwired to 00.
8	0h RO	**Master Data Pariy Error Detected (DPD):** Not implemented, hardwired to 0.
7	0h RO	**Fast Back-to-Back Capable (FBC):** Not implemented, hardwired to 0.
6	0h RO	**Reserved (RSVD_6_6):** Reserved
5	0h RO	**66 MHz Capable (C66):** Not implemented, hardwired to 0.
4	1h RO	**Capabilities List (CL):** Indicates the presence of a capabilities list, hardwired to 1.
3	0h RO/V	**Interrupt Status (IS):** Reflects the state of the INTx# signal at the input of the enable/disable circuit. This bit is a 1 when the INTx# is asserted. This bit is a 0 after the interrupt is cleared (independent of the state of the Interrupt Disable bit in the command register). Note that this bit is not set by an MSI.
2:0	0h RO	**Reserved (RSVD_2_0):** Reserved

24.4 Revision ID And Class Code (HECI3_RID_CC) — Offset 8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + 8h	7800000 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	7h RO	**Base Class Code (BCC):** Indicates the base class code of the HECI host controller device.
23:16	80h RO	**Sub Class Code (SCC):** Indicates the sub class code of the HECI host controller device.
15:8	0h RO	**Programming Interface (PI):** Indicates the programming interface of the HECI host controller device.
7:0	0h RO/V	**Revision ID (RID):** Indicates stepping of the HECI host controller. This field is set by a special IOSF SB message (SetIDValue).

24.5 Cache Line Size (HECI3_CLS) — Offset Ch

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:2] + Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Cache Line Size (CLS):** Not implemented, hardwired to 0.

24.6 Master Latency Timer (HECI3_MLT) — Offset Dh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:2] + Dh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Master Latency Timer (MLT):** Not implemented, hardwired to 0.

24.7 Header Type (HECI3_HTYPE) — Offset Eh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:2] + Eh	80 h

Bit Range	Default & Access	Field Name (ID): Description
7	1h RO	**Multi-Function Device (MFD):** Indicates the HECI host controller is part of a multi-function device.
6:0	0h RO	**Header Layout (HL):** Indicates that the HECI host controller uses a target device layout.

24.8 Built In Self-Test (HECI3_BIST) — Offset Fh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:2] + Fh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**BIST Capable (BC):** Not implemented, hardwired to 0.
6:0	0h RO	**Reserved (RSVD):** Reserved

24.9 HECI MMIO Base Address Low (HECI3_MMIO_MBAR_LO) — Offset 10h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + 10h	4 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**Base Address Low (BA_LO):** Lower 32 bits of base address of register memory space.
11:4	0h RO	**Memory Size (MS):** This field is hardwired to 0 to indicate that HECI device requests 4KB of memory space.
3	0h RO	**Prefetchable (PF):** Indicates that this range is not pre-fetchable.
2:1	2h RO	**Type (TP):** Indicates that this range can be mapped anywhere in 64-bit address space.
0	0h RO	**Resource Type Indicator (RTE):** Indicates a request for register memory space.

24.10 HECI MMIO Base Address High (HECI3_MMIO_MBAR_HI) — Offset 14h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + 14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Base Address High (BA_HI):** Upper 32 bits of base address of register memory space.

24.11 Sub System Identifiers (HECI3_SS) — Offset 2Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**Subsystem ID (SSID):** Indicates the subsystem identifier. This field should be programmed by BIOS during boot-up. Once written, this register becomes Read Only.
15:0	0h RW/O	**Subsystem Vendor ID (SSVID):** Indicates the subsystem vendor identifier. This field should be programmed by BIOS during boot- up. Once written, this register becomes Read Only.

24.12 Capabilities Pointer (HECI3_CAP) — Offset 34h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + 34h	50 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD):** Reserved
7:0	50h RO	**Capability Pointer (CP):** Indicates the first capability pointer offset. It points to the PCI power management capability offset.

24.13 Interrupt Information (HECI3_INTR) — Offset 3Ch

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:2] + 3Ch	100 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	1h RO/V	**Interrupt Pin (IPIN):** This field indicates the virtual interrupt pin the HECI host controller uses. A value of 0x1/0x2/0x3/0x4 indicates that this function implements legacy interrupt on INTA/INTB/INTC/INTD, respectively. When legacy interrupt is not used this register should be set to 0x0. Programmed by BIOS over IOSF SB through corresponding Private CR register.
7:0	0h RW	**Interrupt Line (ILINE):** Software written value to indicate which interrupt line (vector) the interrupt is connected to. No hardware action is taken on this field.

24.14 Minimum Grant (HECI3_MGNT) — Offset 3Eh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:2] + 3Eh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**Grant (GNT):** Not implemented, hardwired to 0.

24.15 Maximum Latency (HECI3_MLAT) — Offset 3Fh

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:2] + 3Fh	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**LAT:** Not implemented, hardwired to 0.

24.16 Host Firmware Status (HECI3_HFS) — Offset 40h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + 40h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Firmware Status Host Access (FS_HA):** Indicates current status of the firmware for the HECI controller. This field is the host's read only access to the FS field in the CSE Firmware Status register.

24.17 General Status Shadow 1 (HECI3_GS_SHDW1) — Offset 48h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + 48h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 1 (GSS1):** This field is host side shadow of CSE General Status 1 (CSE_GS1).

24.18 Host General Status (HECI3_H_GS1) — Offset 4Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + 4Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Host General Status (H_GS1):** General status of Host. This field is not used by hardware.

24.19 PCI Power Management Capability ID (HECI3_PID) — Offset 50h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:2] + 50h	8C01 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	8Ch RO	**NEXT:** Indicates the location of the next capability item in the list. This is the Message Signaled Interrupts capability.
7:0	1h RO	**Cap ID (CID):** Indicates that this pointer is a PCI power management.

24.20 PCI Power Management Capabilities (HECI3_PC) — Offset 52h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:2] + 52h	4003 h

Bit Range	Default & Access	Field Name (ID): Description
15:11	8h RO	**PME Support (PSUP):** Indicates the states that can generate PME#. HECI can assert PME# from D3hot only.
10	0h RO	**D2 Support (D2S):** The D2 state is not supported for the HECI host controller.
9	0h RO	**D1 Support (D1S):** The D1 state is not supported for the HECI host controller.
8:6	0h RO	**Aux Current (AUXC):** Reports the maximum Suspend well current required when in the D3COLD state. Value of 0 is reported.
5	0h RO	**Device Specific Initialization (DSI):** Indicates whether device-specific initialization is required.
4	0h RO	**Reserved (RSVD):** Reserved
3	0h RO	**PME Clock (PMEC):** Indicates that PCI clock is not required to generate PME#.
2:0	3h RO	**Version (VS):** Indicates support for Revision 1.2 of the PCI Power Management Specification.

24.21 PCI Power Management Control And Status (HECI3_PMCS) — Offset 54h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:2] + 54h	8 h

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C/V	**PME Status (PMES):** The PME Status bit in HECI space can be set to '1' by CSE FW performing a write into CSE side register to set PMES. This bit is cleared by host CPU writing a '1' to it. CSE cannot clear this bit. Host CPU writes with value '0' have no effect on this bit.
14:9	0h RO	**Reserved (RSVD_14_9):** Reserved
8	0h RW	**PME Enable (PMEE):** When set, PME_assert and PME_deassert messages are sent over Sideband to PMC based on the PMES bit. When reset these messages may not be sent.
7:4	0h RO	**Reserved (RSVD_7_4):** Reserved
3	1h RO	**No Soft Reset (NSR):** This bit indicates that when the HECI host controller is transitioning from D3hot to D0 due to power state command, it does not perform an internal reset.
2	0h RO	**Reserved (RSVD_2_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Power State (PS):** This field is used both to determine the current power state of the HECI host controller and to set a new power state. The values are: 00 - D0 state 11 - D3HOT state. The D1 and D2 states are not supported for this HECI host controller. If software attempts to write an unsupported, optional state to this field, the write operation must complete normally on the bus; however, the data is discarded and no state change occurs. When in the D3HOT state, the HBA's configuration space is available, but the registers memory space is not. Additionally, interrupts are blocked. This field is visible to firmware through the H_PCI_CSR register, and changes to this field may be configured by the H_PCI_CSR register to generate an MSI to mIA.

24.22 General Status Shadow 2 (HECI3_GS_SHDW2) — Offset 60h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + 60h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 2 (GSS2):** This field is host side shadow of CSE General Status 2 (CSE_GS2).

24.23 General Status Shadow 3 (HECI3_GS_SHDW3) — Offset 64h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + 64h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 3 (GSS3):** This field is host side shadow of CSE General Status 3 (CSE_GS3).

24.24 General Status Shadow 4 (HECI3_GS_SHDW4) — Offset 68h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + 68h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 4 (GSS4):** This field is host side shadow of CSE General Status 4 (CSE_GS4).

24.25 General Status Shadow 5 (HECI3_GS_SHDW5) — Offset 6Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + 6Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**General Status Shadow 5 (GSS5):** This field is host side shadow of CSE General Status 5 (CSE_GS5).

24.26 Host General Status 2 (HECI3_H_GS2) — Offset 70h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + 70h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Host General Status 2 (H_GS2):** General status of Host. This field is not used by hardware.

24.27 Host General Status 3 (HECI3_H_GS3) — Offset 74h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + 74h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Host General Status 3 (H_GS3):** General status of Host. This field is not used by hardware.

24.28 Message Signaled Interrupt Identifiers (HECI3_MID) — Offset 8Ch

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:2] + 8Ch	A405 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	A4h RO	**NEXT:** Indicates the location of the next capability item in the list. This is the DevIdle conventional PCI vendor specific capability.
7:0	5h RO	**Capability ID (CID):** Indicates MSI.

24.29 Message Signaled Interrupt Message Control (HECI3_MC) — Offset 8Eh

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:2] + 8Eh	80 h

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RO	**Reserved (RSVD):** Reserved
7	1h RO	**64 Bit Address Capable (C64):** Specifies whether capable of generating 64-bit messages.
6:4	0h RO	**Multiple Message Enable (MME):** Not implemented, hardwired to 0.
3:1	0h RO	**Multiple Message Capable (MMC):** Not implemented, hardwired to 0.
0	0h RW	**MSI Enable (MSIE):** If set, MSI is enabled and INTx virtual wires are not used to generate interrupts.

24.30 Message Signaled Interrupt Message Address (HECI3_MA) — Offset 90h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + 90h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**ADDR:** Lower 32 bits of the system specified message address, always DW aligned.
1:0	0h RO	**Reserved (RSVD):** Reserved

24.31 Message Signaled Interrupt Upper Address (HECI3_MUA) — Offset 94h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + 94h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Upper Address (UADDR):** Upper 32 bits of the system specified message address. This register is optional and only implemented if MC.C64=1.

24.32 Message Signaled Interrupt Message Data (HECI3_MD) — Offset 98h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:2] + 98h	0 h

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RW	**DATA:** This 16-bit field is programmed by system software if MSI is enabled. Its content is driven onto the FSB during the data phase of the MSI memory write transaction.

24.33 HECI Interrupt Delivery Mode (HECI3_HIDM) — Offset A0h

Type	Size	Offset	Default
CFG	8 bit	[B:0, D:15, F:2] + A0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
7:3	0h RO	**Reserved (RSVD):** Reserved
2	0h RW/1S/V	**HIDM Lock (HIDM_L):** Writing 1 to this bit locks the HIDM field.
1:0	0h RW/L	**HECI Interrupt Delivery Mode (HIDM):** These bits control what type of interrupt the HECI will send when CSE FW writes to set the CSE_IG bit. They are interpreted as follows: 00 - Generate Legacy or MSI interrupt; 01 - Generate SCI; 10 - Generate SMI; This field may be locked by writing 1 to HIDM_L bit.

24.34 Vendor Specific Capability Register (HECI3_VSCR) — Offset A4h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + A4h	F0140009 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RO	**Vendor-Specific Capability ID (VSID):** A value of 4'hF in this 4-bit field identifies this Vendor Specific Capability as an Extended Capability, which uses a VSEC 16-bit Extended Capability ID in the subsequent four bytes. This field allows software to differentiate this capability from other Vendor Specific capabilities.
27:24	0h RO	**Vendor Specific Capability Revision (VSREV):** For a VSID of 4'hF, this field is reserved 4'h0.
23:16	14h RO	**Vendor Specific Capability Length (VSLEN):** This field indicates the number of bytes of this Vendor Specific capability as required by the PCI spec inclusive of the CapID and Next Pointer. It has the value of 8'h14 for the DevIdle Capability.
15:8	0h RO	**NEXT:** Pointer to the configuration offset of the next Capability item. It is 8'h00 since it is the final item in the Capability list.
7:0	9h RO	**Capability ID (CAPID):** The value of 8h09 in this field indicates a Vendor Specific capability.

24.35 Vendor Specific Extended Capability Register (HECI3_VSEC) — Offset A8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + A8h	1400010 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	14h RO	**Vendor Specific Extended Capability Length (VSECLEN):** This field indicates the number of bytes of this Vendor Specific capability inclusive of the 1st DW which includes the capability ID and Next pointer. For DevIdle, this is 12'h14.
19:16	0h RO	**Vendor Specific Extended Capability Revision (VSECREV):** For this revision of DevIdle, this field is 4h0.
15:0	10h RO	**Vendor Specific Extended Capability ID (NEXT):** DevIdle has been assigned the Intel VSEC ID of 16h10.

24.36 SW LTR Pointer Register (HECI3_SWLTRPTR) — Offset ACh

Software Latency Tolerance Reporting (LTR) Pointer Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + ACh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**SW LTR Update MMIO Offset Location (SWLTRLOC):** This register contains the location pointer to the SW LTR register in MMIO space, as an offset from the specified BAR. The value of this register is not valid, if the Valid bit is not set.
3:1	0h RO	**Base Address Register Number (BARNUM):** Contains the 0's based AR Number of the BAR which contains the location of the SW LTR MMIO register. When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number. Bar 0 is the 1st BAR. The value of this register is not valid, if the Valid bit is not set.
0	0h RO	**VALID:** Set to 1'b1 to indicate that the function has implemented a SW LTR register as specified in the DevIdle that can be located using the SWLTRLOC register and BARNUM. Set to 1'b0 to indicate that the function has not implemented a SW LTR register that is compliant to the DevIdle definition. This could be because the function has not implemented SW LTR at all, or has a device specific version of SW LTR.

24.37 Device Idle Pointer Register (HECI3_DEVIDLEPTR) — Offset B0h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + B0h	8001 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	800h RO	**Device Idle MMIO Offset Location (DEVIDLELOC):** This register contains the location pointer to the DevIdle register in MMIO space, as an offset from the specified BAR. The value of this register is not valid, if the Valid bit is not set.
3:1	0h RO	**Base Address Register Number (BARNUM):** Contains the 0's based AR Number of the BAR which contains the location of the SW LTR MMIO register. When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR Number. Bar 0 is the 1st BAR. The value of this register is not valid, if the Valid bit is not set.
0	1h RO	**VALID:** Set to 1'b1 to indicate that the function has implemented a SW LTR register as specified DevIdle that can be located using the DEVIDLELOC register and BARNUM. Set to 1'b0 to indicate that the function has not implemented a DevIdle register that is compliant to the DevIdle definition. This could be because the function has not implemented SW LTR at all, or has a device specific version of device idle.

24.38 Device Idle Power On Latency (HECI3_DEVIDLEPOL) — Offset B4h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:2] + B4h	D38 h

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	**Reserved (RSVD_15_13):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
12:10	3h RW/O	**Power On Latency Scale (POLS):** Latency Scale multiplier: 010: 1us 011: 32us All other settings are reserved (not enforced by HW). The value of this register is multiplied with the Power On Latency Value (POLV) to provide the DevIdle power on exit latency multiplier scale from 1us to 32ms. This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value. This register is undefined if the device idle valid bit is 1'b0.
9:0	138h RW/O	**Power On Latency Value (POLV):** 10-bit value that is multiplied by the Power On Latency Scale. A value of 0 indicates a power on latency of less than 1us. This register is defined to be RWO (read-write-once) to allow BIOS to set the initial value. This register is undefined if the device idle valid bit is 1'b0.

24.39 DevIdle Power Control Enabled Register (HECI3_PWRCTRLEN) — Offset B6h

Type	Size	Offset	Default
CFG	16 bit	[B:0, D:15, F:2] + B6h	E h

Bit Range	Default & Access	Field Name (ID): Description
15:6	0h RO	**Reserved (RSVD_15_6):** Reserved
5	0h RO	**Hardware Autonomous Enable (HAE):** CSE does not support autonomous power gating on idle.
4	0h RO	**Reserved (RSVD_4_4):** Reserved
3	1h RO	**Sleep Enable (SE):** CSE always asserts the sleep signal during power gating.
2	1h RO	**D3-Hot Enable (D3HEN):** CSE could power gate when idle and the PMCSR[1:0] register in the function =2'b11 (D3).
1	1h RO	**DEVIDLE Enable (DEVIDLEN):** CSE could power gate when idle and the DevIdle register (DevIdleC[2] = 1'b1) is set.
0	0h RO	**PMC Request Enable (PMCRE):** CSE will not power gate due to pmc_ip_sw_pg_req_b = LOW.

24.40 Manufacturer's ID (HECI3_MANID) — Offset F8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:15, F:2] + F8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD_31_28):** Reserved
27:24	0h RO/V	**Dot Portion of Process ID (DPOP):** Indicates the dot process. This field is set by a special IOSF SB message (SetIDValue).

Bit Range	Default & Access	Field Name (ID): Description
23:16	0h RO/V	**Manufacturing Stepping ID (MSID):** This field is incremented for each stepping of the part. Note that this field can be used by software to differentiate steppings when Revision ID may not change. This field is set by a special IOSF SB message (SetIDValue).
15:8	0h RO/V	**Manufacturing ID (MID):** This field is set by a special IOSF SB message (SetIDValue).
7:0	0h RO/V	**Process Portion of Process ID (PPOP):** Indicates the process. The dot portion of the process is reflected in bits [27:24]. Note that this field can be used by software to differentiate steppings when Revision ID may not change. This field is set by a special IOSF SB message (SetIDValue).

§ §

25 HD Audio Registers

This chapter documents the registers in Bus: 0, Device 14, Function 0.

25.1 VID_DID — Offset 0h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:14, F:0] + 0h	8C208086 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	8C20h RO/V	**Device ID (DID):** Indicates the device number assigned by the SIG. Bits(2:0) of the DID is controlled by fuse. IOSF Sideband Set ID Value message initializes bits 15:7 of this register value, whilst bits 6:0 of this register value are hardcoded. See the Global Device ID table in Register and Memory Mappings CSpec chapter for the DID value.
15:0	8086h RO	**Vendor ID (VID):** Indicates that Intel is the vendor.

25.2 CMD_STS — Offset 4h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:14, F:0] + 4h	100000 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Detected Parity Error (DPE):** Not implemented. Hardwired to 0.
30	0h RO	**SERR# Status (SERRS):** Not implemented. Hardwired to 0.
29	0h RW/1C/V	**Received Master Abort (RMA):** If the completion status received from IOSF is UR, this bit is set. SW writes a 1 to this bit to clear it.
28	0h RW/1C/V	**Received Target Abort (RTA):** If the completion status received from IOSF is CA, this bit is set. SW writes a 1 to this bit to clear it.
27	0h RO	**Signaled Target-Abort (STA):** Not implemented. Hardwired to 0.
26:25	0h RO	**DEVSEL# Timing Status (DEVT):** Does not apply. Hardwired to 0.
24	0h RO	**Master Data Parity Error (MDPE):** Not implemented. Hardwired to 0.
23	0h RO	**Fast Back to Back Capable (FBC):** Does not apply. Hardwired to 0.
22	0h RO	**Reserved (RSVD1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
21	0h RO	**66 MHz Capable (C66):** Does not apply. Hardwired to 0.
20	1h RO	**Capabilities List Exists (CLIST):** Indicates that the controller contains a capabilities pointer list. The first item is pointed to by looking at configuration offset 34h.
19	0h RO/V	**Interrupt Status (IS):** Reflects the state of the INTx# signal at the input of the enable/disable circuit. This bit is a 1 when the INTx# is asserted. This bit is a 0 after the interrupt is cleared (independent of the state of the Interrupt Disable bit in the command register). Note that this bit is not set by an MSI.
18:16	0h RO	**Reserved (RSVD0):** Reserved
15:11	0h RO	**Reserved (RSVD):** Reserved
10	0h RW	**Interrupt Disable (ID):** Enables the device to assert an INTx#. When set, the Intel(r) HD Audio controller' s INTx# signal will be de-asserted. When cleared, the INTx# signal may be asserted. Note that this bit does not affect the generation of MSIs.
9	0h RO	**Fast Back to Back Enable (FBE):** Not implemented. Hardwired to 0.
8	0h RW	**SERR Enable (SEN):** Functionality not implemented. This bit is R/W to pass PCIe compliance testing.
7	0h RO	**Wait Cycle Control (WCC):** Not implemented. Hardwired to 0.
6	0h RW	**Parity Error Response (PER):** Functionality not implemented. This bit is R/W to pass PCIe compliance testing.
5	0h RO	**VGA Palette Snoop (VPS):** Not implemented. Hardwired to 0.
4	0h RO	**Memory Write and Invalidate Enable (MWI):** Not implemented. Hardwired to 0.
3	0h RO	**Special Cycle Enable (SCE):** Not implemented. Hardwired to 0.
2	0h RW	**Bus Master Enable (BME):** • 1: Enable • 0: Disable Controls standard PCI Express bus mastering capabilities for Memory and IO, reads and writes. Note that this also controls MSI generation since MSI are essentially Memory writes.
1	0h RW	**Memory Space Enable (MSE):** When set, enables memory space accesses to the Intel HD Audio controller.
0	0h RO	**I/O Space (IOS):** The Intel HD Audio controller does not implement IO Space, therefore this bit is hardwired to 0.

25.3 RID_PI_SCC_BCC — Offset 8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:14, F:0] + 8h	4030000 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	4h RO	**Base Class Code (BCC):** This register indicates that the function implements a multimedia device.
23:16	3h RO	**Sub Class Code (SCC):** This indicates the device is an Intel HD Audio audio device, in the context of a multimedia device.
15:8	0h RO	**Programming Interface (PI):** Value assigned to the Intel HD Audio subsystem.
7:0	0h RO/V	**Revision ID (RID):** Indicates the device specific revision identifier. IOSF Sideband Set ID Value message initializes this register value.

25.4 CLS_LT_HTYPE_BIST — Offset Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:14, F:0] + Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Built-in Self Test (BIST):** Not implemented in the Intel HD Audio subsystem. Hardwired to 00h.
23	0h RO	**Multi Function Device (MFD):** Value of 0 indicates a single function device. Value of 1 indicates a multi function device.
22:16	0h RO	**Header Type (HTYPE):** Implements Type 0 Configuration header.
15:8	0h RO	**Latency Timer (LT):** Doesn't apply to PCI Express. RO as 00h if PCI Express. If configured to appear as PCI device, maintain RW for Legacy PCI SW compliance.
7:0	0h RW	**Cache Line Size (CLS):** Doesn't apply to PCI Express. PCI Express spec requires this to be implemented as a R/W register but has no functional impact on the PCH.

25.5 Intel HD Audio Base Lower Address (HDALBA) — Offset 10h

This BAR creates a selected size of memory space to signify the base address (lower 32 bits) of the Intel HD Audio memory mapped configuration registers depending on implementation.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:14, F:0] + 10h	4 h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RW	**Lower Base Address (LBA):** Base address for the Intel HD Audio subsystem's memory mapped configuration registers. 16 Kbytes are requested by hardwiring bits 13:4 to 0.

Bit Range	Default & Access	Field Name (ID): Description
13:4	0h RO	**Reserved (RSVD):** Reserved
3	0h RO	**PREF:** Indicates that this BAR is NOT pre-fetchable.
2:1	2h RO	**Address Range (ADDRNG):** Indicates that this BAR can be located anywhere in 64-bit address space.
0	0h RO	**Space Type (SPTYP):** Indicates that this BAR is located in memory space.

25.6 Intel HD Audio Base Upper Address (HDAUBA) — Offset 14h

This BAR creates a selected size of memory space to signify the base address (upper 32 bits) of the Intel HD Audio memory mapped configuration registers, depending on implementation.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:14, F:0] + 14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Intel HD Audio Upper Base Address (UBA):** Upper 32 bits of the Base address for the Intel(r) HD Audio controller's memory mapped configuration registers.

25.7 Shadowed PCI Configuration Lower Base Address (SPCLBA) — Offset 18h

This BAR creates 4 Kbytes of memory space to signify the base address (lower 32 bits) of the shadowed PCI configuration when used as an ACPI device.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:14, F:0] + 18h	4 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Lower Base Address (LBA):** Base address for the PCI Configuration register shadowed to memory mapped. 4 KB is requested by hardwiring bits 11:4 to 0.
11:4	0h RO	**Reserved (RSVD):** Reserved
3	0h RO	**PREF:** Indicates that this BAR is NOT pre-fetchable.
2:1	2h RO/V	**Address Range (ADDRNG):** Indicates that this BAR can be located anywhere in 64-bit address space.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**Space Type (SPTYP):** Indicates that this BAR is located in memory space.

25.8 Shadowed PCI Configuration Upper Base Address (SPCUBA) — Offset 1Ch

This BAR creates 4 Kbytes of memory space to signify the base address (upper 32 bits) of the shadowed PCI configuration when used as an ACPI device.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:14, F:0] + 1Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Upper Base Address (UBA):** Upper 32 bits of the Base address for the PCI Configuration register shadowed to memory mapped.

25.9 Audio DSP Lower Base Address (ADSPLBA) — Offset 20h

This BAR creates a selected size of memory space to signify the base address (lower 32 bits) of the Audio DSP memory mapped configuration registers depending on implementation. The number of LBA bits in this register is depending on the size of the memory window implemented, represented as x in the register table. The x value is determined by the parameter HDMBABC.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:14, F:0] + 20h	4 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW	**Lower Base Address (LBA):** Base address for the Audio DSP memory mapped configuration registers.
19:4	0h RO	**Reserved (RSVD):** Reserved
3	0h RO	**PREF:** Indicates that this BAR is NOT pre-fetchable.
2:1	2h RO	**Address Range (ADDRNG):** Indicates that this BAR can be located anywhere in 64-bit address space.
0	0h RO	**Space Type (SPTYP):** Indicates that this BAR is located in memory space.

25.10 Audio DSP Upper Base Address (ADSPUBA) — Offset 24h

This BAR creates a selected size of memory space to signify the base address (lower 32 bits) of the Audio DSP memory-mapped configuration registers, depending on implementation.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:14, F:0] + 24h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Upper Base Address (UBA):** Upper 32 bits of the Base address for the Audio DSP memory mapped configuration registers.

25.11 SVID_SID — Offset 2Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:14, F:0] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Subsystem Identifier (SID):** These RW bits have no functionality.
15:0	0h RO	**Subsystem Vendor Identifier (SVID):** These RW bits have no functionality.

25.12 INTLN_INTPN — Offset 3Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:14, F:0] + 3Ch	100 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVDP1):** Reserved
23:16	0h RO	**Reserved (RSVDP0):** Reserved
15:12	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
11:8	1h RO	**Interrupt Pin (INTPN):** Identifies the interrupt pin the function uses. • 0h: No interrupt pin • 1h: INTA • 2h: INTB • 3h: INTC • 4h: INTD • 5h Fh: reserved
7:0	0h RW	**Interrupt Line (INTLN):** Hardware does not use this field directly. It is used to communicate to software the interrupt line that the interrupt pin is connected to.

25.13 PID_PC — Offset 50h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:14, F:0] + 50h	C8436001 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	19h RO	**PME_Support (PMES):** Indicates PME# can be generated from D3 and D0 states. Programmable by BIOS for the option to declare SUS well wake is supported or not: 19h (SUS well wake supported) or 09h (SUS well wake not supported).
26	0h RO	**D2_Support (D2S):** The D2 state is not supported.
25	0h RO	**D1_Support (D1S):** The D1 state is not supported.
24:22	1h RO	**Aux_Current (AC):** Reports 55 mA maximum Suspend well current required when in the D3cold state. P rogrammable by BIOS for the option to declare SUS well wake is supported or not: 001b (SUS well wake supported) or 000b (SUS well wake not supported).
21	0h RO	**Device Specific Initialization (DSI):** Indicates that no device-specific initialization is required.
20	0h RO	**Reserved (RSVD):** Reserved
19	0h RO	**PME Clock (PMEC):** Does not apply. Hardwired to 0.
18:16	3h RO	**Version (VS):** Indicates support for Revision 1.2 of the PCI Power Management Specification. Programmable by BIOS to older revision if compatibility issue is found.
15:8	60h RO	**NEXT:** Points to the next capability structure (MSI).
7:0	1h RO	**CAP:** Indicates that this pointer is a PCI power management capability.

25.14 Power Management Control And Status (PCS) — Offset 54h

This register is used to manage the PCI functions power management state as well as to enable/monitor PMEs. PMES and PMEE bits reside in Resume well, and reset by resume reset.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:14, F:0] + 54h	8 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Data (DT):** Does not apply. Hardwired to 0.
23	0h RO	**Bus Power/Clock Control Enable (BPCCE):** Does not apply. Hardwired to 0.
22	0h RO	**B2/B3 Support (B23):** Does not apply. Hardwired to 0.
21:16	0h RO	**Reserved (RSVD3):** Reserved
15	0h RW/1C/V	**PME Status (PMES):** This bit is set when the Intel HD Audio controller would normally assert the PME# signal independent of the state of the PME bit. This bit is cleared on a power-on reset. Software must not make assumptions about the reset state of this bit and must set it appropriately.
14:9	0h RO	**Reserved (RSVD2):** Reserved
8	0h RW	**PME Enable (PMEE):** When set, and if corresponding PMES is also set, the Intel HD Audio subsystem send PME to PMC. This bit is cleared on a power-on reset. Software must not make assumptions about the reset state of this bit and must set it appropriately.
7:4	0h RO	**Reserved (RSVD1):** Reserved
3	1h RO	**No Soft Reset (NSR):** When set (1), this bit indicates that devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset. Configuration Context is preserved. Upon transition from the D3hot to the D0 Initialized state, no additional operating system intervention is required to preserve Configuration Context beyond writing the PowerState bits. When clear (0), devices do perform an internal reset upon transitioning from D3hot to D0 via software control of the PowerState bits. Configuration Context is lost when performing the soft reset. Upon transition from the D3hot to the D0 state, full reinitialization sequence is needed to return the device to D0 Initialized. Regardless of this bit, devices that transition from D3hot to D0 by a system or bus segment reset will return to the device state D0 Uninitialized with only PME context preserved if PME is supported and enabled.
2	0h RO	**Reserved (RSVD0):** Reserved
1:0	0h RW	**Power State (PS):** This field is used both to determine the current power state of the Intel HD Audio subsystem and to set a new power state. The values are: • 00: D0 state • 1 1: D3HOT state If software attempts to write a value of 10b or 01b in to this field, the write operation must complete normally, however, the data is discarded and no state change occurs. When in the D3HOT states, the Intel HD Audio subsystem's configuration space is available, but the I/O and memory spaces are not. Additionally, interrupts are blocked. When software changes this value from the D3HOT state to the D0 state, an internal warm (soft) reset is generated, and software must re-initialize the function.

25.15 MID_MMC — Offset 60h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:14, F:0] + 60h	807005 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23	1h RO	**64b Address Capability (ADD64):** Indicates the ability to generate a 64-bit message address.
22:20	0h RO	**Multiple Message Enable (MME):** Normally this is a R/W register. However, since only 1 message is supported, these bits are hardwired to 000 = 1 message.
19:17	0h RO	**Multiple Message Capable (MMC):** Hardwired to 0 indicating request for 1 message.
16	0h RW	**MSI Enable (ME):** If set to 1 an MSI will be generated instead of an INTx#signal. If set to 0, an MSI may not be generated.
15:8	70h RO/V	Default value:00 **NEXT:** This field has a value of 00h to indicate that this is the last capability structure in the list.
7:0	5h RO	**CAP:** Indicates that this pointer is a MSI capability.

25.16 MSI Message Lower Address (MMLA) — Offset 64h

This register specifies the MSI message address (lower 32 bits).

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:14, F:0] + 64h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**MSI Message Lower Address (MMLA):** Lower Address used for MSI Message.
1:0	0h RO	**Reserved (RSVD):** Reserved

25.17 MSI Message Upper Address (MMUA) — Offset 68h

This register specifies the MSI message address (Upper 32 bits).

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:14, F:0] + 68h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**MSI Message Upper Address (MMUA):** Upper 32 bits of address used for MSIMessage.

25.18 MSI Message Data (MMD) — Offset 6Ch

This register specifies the MSI message data.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:14, F:0] + 6Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVDP1):** Reserved
23:16	0h RO	**Reserved (RSVDP0):** Reserved
15:0	0h RW	**MSI Message Data (MMD):** Data used for MSI Message.

25.19 (GCAP_VMIN_VMAJ)—Offset 0h

This is a BAR0 MMIO Register**.**

AccessMethod

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 1006701h

Bit Range	Default & Access	Field Name (ID): Description
31:24	1h RO	**Major Version (VMAJ):** Indicates the Intel HD Audio controller supports major revision number 1 of the Intel HD Audio specification.
23:16	0h RO	**Minor Version (VMIN):** Indicates the Intel HD Audio controller supports minor revision number 00h of the Intel HD Audio specification.

Bit Range	Default & Access	Field Name (ID): Description
15:12	6h RO	**Number of Output Streams Supported (OSS):** 0100b indicates that the Intel HD Audio controller supports four output streams. [p]Reset value is hardcoded to parameter HSTOSC.
11:8	7h RO	**Number of Input Streams Supported (ISS):** 0100b indicates that the Intel HD Audio controller supports four input streams. [p]Reset value is hardcoded to parameter HSTISC.
7:3	0h RO	**Number of Bidirectional Streams Supported (BSS):** 00000b indicates that the Intel HD Audio controller supports 0 bidirectional streams.
2:1	0h RO	**Number of Serial Data Out Signals (NSDO):** 00b indicates that the Intel HD Audio controller supports one Serial Data Output signal. For the case in which multiple link segments are supported, this field indicates the number of SDOs for link 0.
0	1h RO	**64 Bit Address Supported (ADD64OK):** A 1 indicates that the Intel HD Audio controller supports 64 bit addressing for BDL addresses, data buffer addresses, and command buffer addresses.

25.20 (OUTPAY_INPAY)—Offset 4h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 1D003Ch

Bit Range	Default & Access	Field Name (ID): Description
31:16	1Dh RO	**Input Payload Capability (INPAY):** Indicates the total input payload available on the link. This does not include bandwidth used for response. This measurement is in 16-bit word quantities per 48 kHz frame. The default link clock speed of 24.000 MHz provides 500 bits per frame, or 31.25 words in total. 36 bits are used for response, leaving 29 words for data payload. Note that this value does not reflect any bandwidth increase due to support for multiple SDI lines. • 00h: 0 word • 01h: 1 word payload • ... • FFh: 255h word payload Note: In the event that multiple links are supported (GCAP2.LCOUNT) 0), the payload advertised will be the lowest common denominator offered by the default clock frequency on each link.

Bit Range	Default & Access	Field Name (ID): Description
15:0	3Ch RO	**Output Payload Capability (OUTPAY):** Indicates the total output payload available on the link. This does not include bandwidth used for command and control. This measurement is in 16-bit word quantities per 48 kHz frame. The default link clock speed of 24.000 MHz (the data is double pumped) provides 1000 bits per frame, or 62.5 words in total. 40 bits are used for command and control, leaving 60 words available for data payload. Note that this value does not reflect any bandwidth increase due to support for multiple SDO lines. • 00h: 0 word • 01h: 1 word payload • ... • FFh: 255h word payload Note: In the event that multiple links are supported (GCAP2.LCOUNT) 0), the payload advertised will be the lowest common denominator offered by the default clock frequency on each link.

25.21 Global Control (GCTL)—Offset 8h

CRSTB bit is not affected by controller reset.

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Reserved (RSVD1):** Reserved.
8	0h RW	**Accept Unsolicited Response Enable (UNSOL):** If UNSOL is a 1, Unsolicited Responses from the codecs are accepted by the controller and placed into the Response Input Ring Buffer. If UNSOL is a 0, unsolicited responses are not accepted, and dropped on the floor.
7:2	0h RO	**Reserved (RSVD2):** Reserved.
1	0h RW/1S/V	**Flush Control (FCNTRL):** Writing a 1 to this bit initiates a flush. When the flush completion is received by the controller, hardware sets the Flush Status bit and clears this Flush Control bit. Before a flush cycle is initiated, the DMA Position Buffer must be programmed with a valid memory address by software, but the DMA Position Buffer bit 0 need not be set to enable the position reporting mechanism. Also, all streams must be stopped (the associated RUN bit must be 0). When the flush is initiated, the controller will flush pipelines to memory to guarantee that the hardware is ready to transition to a D3 state. Setting this bit is not a critical step in the power state transition if the content of the FIFOs is not critical.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW/V	**Controller Reset# (CRSTB):** values. After the hardware has completed sequencing into the reset state, it will report a 0 in this bit. Software must read a 0 from this bit to verify that the controller is in reset. Writing a 1 to this bit causes the controller to exit its reset state and deassert the Intel HD Audio link RESET# signal. Software is responsible for setting/clearing this bit such that the minimum Intel HD Audio link RESET# signal assertion pulse width specification is met. When the controller hardware is ready to begin operation, it will report a 1 in this bit. Software must read a 1 from this bit before accessing any controller registers. The CRST# bit defaults to a 0 after hardware reset, therefore software needs to write a 1 to this bit to begin operation. Note that the CORB/RIRB RUN bits and all Stream RUN bits must be verified cleared to zero before CRST# is written to 0 (asserted) in order to assure a clean re-start. When setting or clearing CRST#, software must ensure that minimum link timing requirements (minimum RESET# assertion time, etc.) are met. When CRST# is 0 indicating that the controller is in reset, writes to all Intel HD Audio memory mapped registers are ignored as if the device is not present. The only exception is the Global Control register containing the CRST# bit itself. The Global Control register is write-able as a DWord, Word, or Byte even when CRST# is 0 if the byte enable for the byte containing the CRST# bit (Byte Enable 0) is active. If Byte Enable 0 is not active, writes to the Global Control register will be ignored when CRST# is 0. When CRST# is 0, reads to Intel HD Audio memory mapped registers will return their default value except for registers that are not reset with PLTRST# or on a D3hot to D0 transition.

25.22 (WAKEEN_WAKESTS)—Offset Ch

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD1_1):** Reserved.
18:16	0h RW/1C/V	**SDIN State Change Status Flags (WAKESTS):** Flag bits that indicate which SDI signal(s) received a State Change event. The bits are cleared by writing 1's to them. These bits are cleared on a power-on reset. Software must not make assumptions about the reset state of these bits and must set them appropriately.

Bit Range	Default & Access	Field Name (ID): Description
15:3	0h RO	**Reserved (RSVD1):** Reserved.
2:0	0h RW	**SDIN Wake Enable Flags (WAKEEN):** Bits which control which SDI signal(s) may generate a wake event. A 1 bit in the bit mask indicates that the associated SDIN signal is enabled to generate a wake. These bits are cleared on a power-on reset. Software must not make assumptions about the reset state of these bits and must set them appropriately.[/ p]

25.23 (GSTS_GCAP2)—Offset 10h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 10000h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RO	**Reserved (RSVD1_1):** Reserved.
17	0h RO	**Dynamic FIFO Limit Change Dapability (DFIFOLCC):** Indicates whether the energy efficient audio FIFOLC operation is static or dynamic. • 0: Static. FIFOLC bit only has effect before RUN bit is set for the first time. • 1: Dynamic. FIFOLC bit has effect before RUN bit is set for the first time, as well as after that.
16	1h RW/V	**Energy Efficient Audio Capability (EEAC):** Indicates whether the energy efficient audio with deeper buffering is supported or not. • 0: Not supported. FIFOL register and FIFOLC bit behave as RO. • 1: Supported. FIFOL register and FIFOLC bit behave as RW.
15:2	0h RO	**Reserved (RSVD1):** Reserved.
1	0h RW/1C/V	**Flush Status (FSTS):** This bit is set to a 1 by the hardware to indicate that the flush cycle initiated when the FCNTRL bit was set has completed. Software must write a 1to clear this bit before the next time FCNTRL is set.
0	0h RO	**Reserved (RSVD2):** Reserved.

25.24 Linked List Capabilities Header (LLCH)—Offset 14h

This register resides in Primary well (always on), or resides in Primary well (gated) with state retention, and reset by platform reset.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: C00h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVDP1):** Reserved.
23:16	0h RO	**Reserved (RSVDP0):** Reserved.
15:0	C00h RO	**First Capability Pointer (PTR):** This field contains the offset to the first capability structure of the linked list capabilities, or 0000h if no linked list capabilities exist. Point to Multiple Links Capability.

25.25 (OUTSTRMPAY_INSTRMPAY)—Offset 18h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 180030h

Bit Range	Default & Access	Field Name (ID): Description
31:16	18h RO	**Input Stream Payload Capability (INSTRMPAY):** Indicates maximum number of words per frame for any single input stream. This measurement is in 16-bit word quantities per 48 kHz frame. 24 Words (48B) is the maximum supported, therefore a value of 18h is reported in this register. Software must ensure that a format which would cause more words per frame than indicated is not programmed into the Input Stream Descriptor register. • 00h: 0 word • 01h: 1 word payload • ... • FFh: 255h word payload
15:0	30h RO	**Output Stream Payload Capability (OUTSTRMPAY):** Indicates maximum number of words per frame for any single output stream. This measurement is in 16 bit word quantities per 48 kHz frame. 48 Words (96B) is the maximum supported, therefore a value of 30h is reported in this register. Software must ensure that a format which would cause more words per frame than indicated is not programmed into the Output Stream Descriptor register. • 00h: 0 word • 01h: 1 word payload • ... • FFh: 255h word payload

25.26 Interrupt Control (INTCTL)—Offset 20h

The Interrupt Status and Control register provides a central point for controlling and monitoring interrupt generation. The SIE (Stream Interrupt Enable) register controls the interrupt mask for each individual Input or Output Stream. Setting a 1 in the appropriate bit allows the particular interrupt source to generate a processor interrupt. The SIS (Stream Interrupt Status) register indicates the current interrupt status of each interrupt source. A 1 indicates that an interrupt is being requested. Note that the state of these bits is independent of the SIE bits; even if the corresponding bit is set to a 0 in the Stream Interrupt Enable register to disable processor interrupt generation, the Status bit may still be set to indicate that stream is requesting service. This can be used by polling software to determine which Streams need attention without incurring system interrupts.
The CIE (Controlle r Interrupt Enable) and CIS (Controller Interrupt Status) control and indicate the status of the general controller interrupt. General controller interrupt sources are to a Response Interrupt, a Response Buffer Overrun, and State Change events. Note that the CIS is independent of the CIE bit; even if the CIE bit is set to a 0 to disable processor interrupt generation, the CIS bit may still be set to indicate that stream is requesting service.
The GIE (Global Interrupt Enable) and GIS (Global Interrupt Status) control and indicate the status of all hardware interrupt sources in the Intel HD Audio controller. If GIS bit is a 1, a processor interrupt is currently being requested. If GIE is a 1, a processor interrupt may be requested; if GIE is a 0, then no processor interrupt may be requested. Note that the GIS is independent of the GIE bit; even if the GIE bit is set to a 0 to disable processor interrupt generation, the GIS bit may still be set to indicate that stream is requesting service.
GIE and CIE bits are not affected by controller reset.
The number of SIE bits in this register is depending on the total number of stream DMA implemented, represented as x in the register table. The x value is determined by the sum of parameter HSTISC and HSTOSC.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Global Interrupt Enable (GIE):** Global bit to enable device interrupt generation. When set to 1 the Intel HD Audio function is enabled to generate an interrupt. This control is in addition to any bits in the bus specific address space, such as the Interrupt Enable bit in the PCI Configuration Space.
30	0h RW	**Controller Interrupt Enable (CIE):** Enables the general interrupt for controller functions. When set to 1 (and GIE is enabled), the controller generates an interrupt when the CIS bit gets set.
29:13	0h RO	**Reserved (RSVD1):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
12:0	0h RW	**Stream Interrupt Enable (SIE):** When set to 1 the individual Streams are enabled to generate an interrupt when the corresponding stream status bits get set. A stream interrupt will be caused as a result of a buffer with IOC=1 in the BDL entry being completed, or as a result of a FIFO error (underrun or overrun) occurring. Control over the generation of each of these sources is in the associated Stream Descriptor. The streams are numbered and the SIE bits assigned sequentially, based on their order in the register set.

25.27 Interrupt Status (INTSTS)—Offset 24h

GIS and CIS bits are not affected by controller reset.
The number of SIS bits in this register is depending on the total number of stream DMA implemented, represented as x in the register table. The x value is determined by the sum of parameter HSTISC and HSTOSC.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Global Interrupt Status (GIS):** This bit is an OR of all of the interrupt status bits in this register and PPSTS register.
30	0h RW/V	**Controller Interrupt Status (CIS):** Status of general controller interrupt. A 1 indicates that an interrupt condition occurred due to a Response Interrupt, a Response Buffer Overrun Interrupt, CORB Memory Error Interrupt, or a SDIN State Change event. The exact cause can be determined by interrogating other registers. Note that a HW interrupt will not be generated unless the corresponding enable bit is set. This bit is an OR of all of the stated interrupt status bits for this register.
29:13	0h RO	**Reserved (RSVD1):** Reserved.
12:0	0h RW/V	**Stream Interrupt Status (SIS):** A 1 indicates that an interrupt condition occurred on the corresponding Stream. Note that a HW interrupt will not be generated unless the corresponding enable bit is set. This bit is an OR of all of an individual stream s interrupt status bits. Th e streams are numbered and the SIS bits assigned sequentially, based on their order in the register set.

25.28 Wall Clock Counter (WALCLK)—Offset 30h

The 32-bit monotonic counter provides a 'wall clock' that can be used by system software to synchronize independent audio controllers. The counter must be implemented.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Wall Clock Counter (WALCLK):** 32-bit counter that is incremented on each link BCLK period and rolls over from FFFF_FFFFh to 0000_0000h. This counter will roll over to zero with a period of approximately 179 seconds. This counter is enabled while the BCLK bit is set to 1. Software uses this counter to synchronize between multiple controllers. Will be reset on controller reset. With the introduction of multiple link segments for the Intel HD Audio controller, and the capability of running each link segment at different clock speed, the BCLK definition for this counter is fixed at 24 MHz equivalent rate always, independent of the physical link clock speed, and reports the link 0 wall clock value.

25.29 Stream Synchronization (SSYNC)—Offset 38h

To synchronize two or more streams the corresponding SSYNC bits for the streams to be synchronized should be set to 1 before the 'RUN' bit for each stream is set.
The RUN bit for the corresponding stream must be set to 1 (and FIFORDY=1) prior to that stream's SSYNC bit being written to 0. To start multiple streams synchronously, the stream sync bits for those streams should be written to 0 at the same time. For all SSYNC bits on output engines that transition from 1 to 0 on the same write, the formatter will deliver a sample over the link in the same 48kHz frame. For all SSYNC bits on input engines that transition from 1 to 0 on the same write, the formatter will take stream data off the link and place it in the FIFO.
If synchronization is not desired, the stream synchronization bits may be left 0, and the stream will simply begin running normally when the stream's 'RUN' bit is set.
In addition to platform reset, FLR, and controller reset, the register is also reset by stream reset.
The number of SSYNC bits in this register is depending on the total number of stream DMA implemented, represented as x in the register table. The x value is determined by the sum of parameter HSTISC and HSTOSC.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:13	0h RO	**Reserved (RSVD1):** Reserved.
12:0	0h RW	**Stream Synchronization Bits (SSYNC):** The Stream Synchronization bits, when set to 1, block data from being sent on or received from the link. Each bit controls the associated Stream Descriptor, bit 0 corresponds to the first Stream Descriptor, etc. To synchronously start a set of DMA engines, the bits in the SSYNC register are first set to a 1. The RUN bits for the associated Stream Descriptors are then set to a 1 to start the DMA engines. When all streams are ready (FIFORDY=1), the associated SSYNC bits can all be set to 0 at the same time, and transmission or reception of bits to or from the link will begin together at the start of the next full link frame. To synchronously stop streams, first the bits are set in the SSYNC register, and then the individual RUN bits in the Stream Descriptors are cleared by software. The streams are numbered and the SSYNC bits assigned sequentially, based on their order in the register set.

25.30 CORB Lower Base Address (CORBLBASE)—Offset 40h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RW	**CORB Lower Base Address (CORBLBASE):** Lower address of the Command Output Ring Buffer, allowing the CORB Base Address to be assigned on any 128-B boundary. This register field must not be written when the DMA engine is running or the DMA transfer may be corrupted.
6:0	0h RO	**CORB Lower Base Unimplemented Bits (RSVD):** Hardwired to 0. This requires the CORB to be allocated with 128-byte granularity to allow for cache line fetch optimizations.

25.31 CORB Upper Base Address (CORBUBASE)—Offset 44h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**CORB Upper Base Address (CORBUBASE):** Upper 32 bits of address of the Command Output Ring Buffer. This register field must not be written when the DMA engine is running or the DMA transfer may be corrupted.

25.32 (CORBWP_CORBRP)—Offset 48h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/V	**CORB Read Pointer Reset (CORBRPRST):** Software writes a 1 to this bit to reset the CORB Read Pointer to 0 and clear any residual pre-fetched commands in the CORB hardware buffer within the Intel Audio controller. The hardware will physically update this bit to 1 when the CORB Pointer reset is complete. Software must read a 1 to verify that the reset completed correctly. Software must clear this bit back to 0 and read back the 0 to verify that the clear completed correctly. The CORB DMA engine must be stopped prior to resetting the Read Pointer or else DMA transfer may be corrupted.
30:24	0h RO	**Reserved (RSVD1_1):** Reserved.
23:16	0h RO/V	**CORB Read Pointer (CORBRP):** Software reads this field to determine how many commands it can write to the CORB without over-running. The value read indicates the CORB Read Pointer offset in Dword granularity. The offset entry read from this field has been successfully fetched by the DMA controller and may be over-written by software. Supports 256 CORB entries (256 x 4B=1KB). This field may be read while the DMA engine is running.
15:8	0h RO	**Reserved (RSVD1):** Reserved.
7:0	0h RW	**CORB Write Pointer (CORBWP):** Software writes the last valid CORB entry offset into this field in Dword granularity. The DMA engine fetches commands from the CORB until the Read Pointer matches the Write Pointer. Supports 256 CORB entries (256 x 4B=1KB). This field may be written while the DMA engine is running.

25.33 (CORBCTL_CORBSTS_CORBSIZE)—Offset 4Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 420000h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVDP0):** Reserved.
23:20	4h RO	**CORB Size Capability (CORBSZCAP):** 0100b indicates that the ICH9 only supports a CORB size of 256 CORB entries (1024B).
19:18	0h RO	**Reserved (RSVD1_2):** Reserved.
17:16	2h RO	**CORB Size (CORBSIZE):** Hardwired to 10b which sets the CORB size to 256 entries (1024B).
15:9	0h RO	**Reserved (RSVD1_1):** Reserved.
8	0h RW/1C/V	**CORB Memory Error Indication (CMEI):** If this status bit is set, the controller has detected an error in the pathway between the controller and memory. This may be an ECC bit error or any other type of detectable data error which renders the command data fetched invalid. Software can clear this bit by writing a 1 to it. However, this type of error leaves the audio subsystem in an unviable state and typically requires CRST#.
7:2	0h RO	**Reserved (RSVD1):** Reserved.
1	0h RW/V	**Enable CORB DMA Engine (CORBRUN):** • 0: DMA Stop • 1: DMA Run After SW writes a 0 to this bit, the HW may not stop immediately. The hardware will physically update the bit to a 0 when the DMA engine is truly stopped. SW must read a 0 from this bit to verify that the DMA is truly stopped.
0	0h RW	**CORB Memory Error Interrupt Enable (CMEIE):** If this bit is set (and GIE and CIE are enabled), the controller will generate an interrupt if the MEI status bit is set.

25.34 RIRB Lower Base Address (RIRBLBASE)—Offset 50h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RW	**RIRB Lower Base Address (RIRBLBASE):** Lower address of the Response Input Ring Buffer, allowing the RIRB Base Address to be assigned on any 128-B boundary. This register field must not be written when the DMA engine is running or the DMA transfer may be corrupted.
6:0	0h RO	**RIRB Lower Base Unimplemented Bits (RSVD):** Hardwired to 0 to force 128-byte buffer alignment for cache line fetch optimizations.

25.35 RIRB Upper Base Address (RIRBUBASE)—Offset 54h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RIRB Upper Base Address (RIRBUBASE):** Upper 32 bits of address of the Command Output Ring Buffer. This register field must not be written when the DMA engine is running or the DMA transfer may be corrupted.

25.36 (RIRBWP_RINTCNT)—Offset 58h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD1_1):** Reserved.
23:16	0h RW	**N Response Interrupt Count (RINTCNT):** • 0000_0001b: 1 Response sent to RIRB • ... • 1111_1111b: 255 Responses sent to RIRB • 0000_0000b: 256 Responses sent to RIRB The DMA engine should be stopped when changing this field or else an interrupt may be lost. [p]Note that each Response occupies 2 Dwords in the RIRB. This is compared to the total number of responses that have been returned, as opposed to the number of frames in which there were responses. If more than one codec responds in one frame, then the count is increased by the number of responses received in the frame.

Bit Range	Default & Access	Field Name (ID): Description
15	0h WO	**RIRB Write Pointer Reset (RIRBWPRST):** Software writes a 1 to this bit to reset the RIRB Write Pointer to 0's. The RIRB DMA engine must be stopped prior to resetting the Write Pointer or else DMA transfer may be corrupted. This bit will always be read as 0.
14:8	0h RO	**Reserved (RSVD1):** Reserved.
7:0	0h RO/V	**RIRB Write Pointer (RIRBWP):** Indicates the last valid RIRB entry written by the DMA controller. Software reads this field to determine how many responses it can read from the RIRB. The value read indicates the RIRB Write Pointer offset in 2 Dword RIRB entry units (since each RIRB entry is 2 Dwords long). Supports up to 256 RIRB entries (256 x 8B=2KB). This field may be read while the DMA engine is running.

25.37 (RIRBCTL_RIRBSTS_RIRBSIZE)—Offset 5Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 420000h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVDP0):** Reserved.
23:20	4h RO	**RIRB Size Capability (RIRBSZCAP):** 0100b indicates that the PCH only supports a RIRB size of 256 RIRB entries (2048B) .
19:18	0h RO	**Reserved (RSVD1_2):** Reserved.
17:16	2h RO	**RIRB Size (RIRBSIZE):** Hardwired to 10b which sets the RIRB size to 256 entries (2048B).
15:11	0h RO	**Reserved (RSVD1_1):** Reserved.
10	0h RW/1C/V	**Response Overrun Interrupt Status (RIRBOIS):** Hardware sets this bit to a 1 when the RIRB DMA engine is not able to write the incoming responses to memory before additional incoming responses overrun the internal FIFO. When the overrun occurs, the hardware will drop the responses which overrun the buffer. An interrupt may be generated if the Response Overrun Interrupt Control bit is set. Note that this status bit is set even if an interrupt is not enabled for this event. Software clears this flag by writing a 1 to this bit.
9	0h RO	**Reserved (RSVD2):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW/1C/V	**Response Interrupt (RINTFL):** Hardware sets this bit to a 1 when an interrupt has been generated after N number of Responses are sent to the RIRB buffer OR when an empty Response slot is encountered on all SDI(x) inputs (whichever occurs first). Note that this status bit is set even if an interrupt is not enabled for this event. Software clears this flag by writing a 1 to this bit.
7:3	0h RO	**Reserved (RSVD1):** Reserved.
2	0h RW	**Response Overrun Interrupt Control (RIRBOIC):** If this bit is set (and GIE and CIE are enabled), the hardware will generate an interrupt when the Response Overrun Interrupt Status bit is set.
1	0h RW/V	**RIRB DMA Enable (RIRBRUN):** • 0: DMA Stop • 1: DMA Run After SW writes a 0 to this bit, the HW may not stop immediately. The hardware will physically update the bit to a 0 when the DMA engine is truly stopped. SW must read a 0 from this bit to verify that the DMA is truly stopped.
0	0h RW	**Response Interrupt Control (RINTCTL):** • 0: Disable Interrupt. • 1: Generate an interrupt (if GIE and CIE are enabled) after N number of Responses are sent to the RIRB buffer OR when an empty Response slot is encountered on all SDI_x inputs (whichever occurs first). The N counter is reset when the interrupt is generated.

25.38 Immediate Command (IC)—Offset 60h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Immediate Command (IC):** The command to be sent to the codec via the Immediate Command mechanism is written to this register. The command stored in this register is sent out over the link during the next available frame after a 1 is written to the ICB bit.

25.39 Immediate Response (IR)—Offset 64h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Immediate Response (IR):** This register contains the response received from a codec resulting from a command sent via the Immediate Command mechanism. If multiple codecs responded in the same frame, there is no guarantee as to which response will be latched. Therefore broadcast-type commands must not be issued via the Immediate Command mechanism.

25.40 Immediate Command Status (ICS)—Offset 68h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVDP1):** Reserved.
23:16	0h RO	**Reserved (RSVDP0):** Reserved.
15:2	0h RO	**Reserved (RSVD1):** Reserved.
1	0h RW/1C/V	**Immediate Result Valid (IRV):** This bit is set to a 1 by hardware when a new response is latched into the IR register. This is a status flag indicating that software may read the response from the Immediate Response register. Software must clear this bit (by writing a one to it) before issuing a new command so that the software may determine when a new response has arrived.
0	0h RW/V	**Immediate Command Busy (ICB):** When this bit as read as a 0 it indicates that a new command may be issued using the Immediate Command mechanism. When this bit transitions from a 0 to a 1 (via software writing a 1), the controller issues the command currently stored in the Immediate Command register to the codec over the link. When the corresponding response is latched into the Immediate Response register, the controller hardware sets the IRV flag and clears the ICB bit back to 0. SW may write this bit to a 0 if the bit fails to return to 0 after a reasonable timeout period. Note that an Immediate Command must not be issued while the CORB/RIRB mechanism is operating, otherwise the responses conflict. This must be enforced by software.

25.41 DMA Position Lower Base Address (DPLBASE)—Offset 70h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RW	**DMA Position Lower Base Address (DPLBASE):** Lower 32 bits of the DMA Position Buffer Base Address. This register field must not be written when any DMA engine is running or the DMA transfer may be corrupted. This same address is used by the Flush Control, and must be programmed with a valid value before the FLCNRTL bit is set.
6:1	0h RO	**DMA Position Lower Base Unimplemented Bits (RSVD):** Hardwired to 0 to force 128 byte buffer alignment for cache line write optimizations.
0	0h RW	**DMA Position Buffer Enable (DPBE):** When this bit is set to a '1', the controller will write the DMA positions of each of the DMA engines to the buffer in main memory periodically (typically once/frame). Software can use this value to know what data in memory is valid data. The controller must guarantee that the values in the DMA Position Buffer that the software can read represent positions in the stream for which valid data exists in the Stream's DMA buffer. This has particular relevance in systems which support isochronous transfer, the stream positions in the software-visible memory buffer must represent stream data which has reached the Global Observation point.

25.42 DMA Position Upper Base Address (DPUBASE)—Offset 74h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**DMA Position Upper Base Address (DPUBASE):** Upper 32 bits of address of the DMA Position Buffer Base Address. This register field must not be written when the DMA engine is running or the DMA transfer may be corrupted.

25.43 (ISD0CTL_ISD0STS)—Offset 80h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 40000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD1_1):** Reserved.
29	0h RO/V	**FIFO Ready (FIFORDY):** This bit defaults to 0 on reset because the FIFO is cleared on a reset. Input Stream: For input streams, the controller hardware will set this bit to 1 when a valid descriptor is loaded and the engine is ready for the RUN bit to be set. Output Stream: For output streams, the controller hardware will set this bit to a 1 while the output DMA FIFO contains enough data to maintain the stream on then link.
28	0h RW/1C/V	**Descriptor Error (DESE):** Indicates that a serious error occurred during the fetch of a descriptor. This could be a result of a Master Abort, a Parity or ECC error on the bus, or any other error which renders the current Buffer Descriptor or Buffer Descriptor List useless. This error is treated as a fatal stream error, as the stream cannot continue running. The RUN bit will be cleared and the stream will stop. Software may attempt to restart the stream engine after addressing the cause of the error and writing a 1 to this bit to clear it.
27	0h RW/1C/V	**FIFO Error (FIFOE):** Set when a FIFO error occurs. Bit is cleared by writing a 1 to this bit position. This bit is set even if an interrupt is not enabled. Input Stream: For an input stream, this indicates a FIFO overrun occurring while the RUN bit is set. When this happens, the FIFO pointers don't increment and the incoming data is not written into the FIFO, thereby being lost. Output Stream: For an output stream, this indicates a FIFO under run when there are still buffers to send. The hardware should not transmit anything on the link for the associated stream if there is not valid data to send.
26	0h RW/1C/V	**Buffer Completion Interrupt Status (BCIS):** This bit is set to 1 by the hardware after the last sample of a buffer has been processed, AND if the Interrupt on Completion (IOC) bit is set in the command byte of the buffer descriptor. It remains active until software clears it by writing a 1 to this bit position.
25:24	0h RO	**Reserved (RSVD2):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
23:20	0h RW	**Stream Number (STRM):** This value reflects the Tag associated with the data being transferred on the link. • 0000=Reserved (Indicates Unused) • 0001=Stream 1 • ... • 1110=Stream 14 • 1111=Stream 15 Input Stream: When an input stream is detected on any of the SDIx signals that match this value, the data samples are loaded into the FIFO associated with this descriptor. Note that while a single SDIx input may contain data from more than one stream number, two different SDIx inputs may not be configured with the same stream number. Output Stream:[/p] When data controlled by this descriptor is sent out over the link, it will have this stream number encoded on the SYNC signal.
19	0h RO	**Bidirectional Direction Control (DIR):** This bit is only meaningful for Bidirectional streams. Therefore this bit is hardwired to 0.
18	1h RO	**Traffic Priority (TP):** Hardwired to 1 indicating that all streams will use VC1 if it is enabled throughout the PCI Express registers.
17:16	0h RO	**Stripe Control (STRIPE):** Input Stream: This field is meaningless for input streams. Output Stream: For output streams it controls the number of SDO signals to stripe data across.
15:6	0h RO	**Reserved (RSVD1):** Reserved.
5	0h RW/V	**FIFO Limit Change (FIFOLC):** Writing a 1 to this bit indicates a new update to the FIFOL register has been made. After the HW has completed sequencing into the new effective FIFO size, it will clear this bit. This bit is RO if GSTS_GCAP2.EEAC = 0. If GSTS_GCAP2.EEAC = 1 and GCAP2.DFIFOLCC = 0 (static), this bit will only have effect when the stream is idle (before the first time RUN bit is set). If GSTS_GCAP2.EEAC = 1 and GCAP2.DFIFOLCC = 1 (dynamic), on top of supporting the static behavior describe above, this bit also has effect when the stream is active (after the first time RUN bit is set).
4	0h RW	**Descriptor Error Interrupt Enable (DEIE):** Controls whether an interrupt is generated when the Descriptor Error Status (DESE) bit is set.
3	0h RW	**FIFO Error Interrupt Enable (FEIE):** This bit controls whether the occurrence of a FIFO error (overrun for input or underrun for output) will cause an interrupt or not. If this bit is not set, bit 3 in the Status register will be set, but the interrupt will not occur. Either way, the samples will be dropped.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Interrupt On Completion Enable (IOCE):** This bit controls whether or not an interrupt occurs when a buffer completes with the IOC bit set in its descriptor. If this bit is not set, bit 2 in the Status register will be set, but the interrupt will not occur
1	0h RW/V	**Stream Run (RUN):** When set to 1 the DMA engine associated with this input stream will be enabled to transfer data in the FIFO to main memory. The SSYNC bit must also be cleared in order for the DMA engine to run. When cleared to 0 the DMA engine associated with this input stream will be disabled. Hardware will report a 0 in this bit when the DMA engine is actually stopped. Software must read a 0 from this bit before modifying related control registers or restarting the DMA engine.
0	0h RW/V	**Stream Reset (SRST):** Writing a 1 causes the corresponding stream to be reset. The Stream Descriptor registers (except the SRST bit itself) and FIFO's for the corresponding stream are reset. After the stream hardware has completed sequencing into the reset state, it will report a 1 in this bit. Software must read a 1 from this bit to verify that the stream is in reset. Writing a 0 causes the corresponding stream to exit reset. When the stream hardware is ready to begin operation, it will report a 0 in this bit. Software must read a 0 from this bit before accessing any of the stream registers. The RUN bit must be cleared before SRST is asserted.

25.44 Input/Output Stream Descriptor x Link Position in Buffer (ISD0LPIB)—Offset 84h

Register for Input/Output Stream Descriptor x Link Position in Buffer.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Link Position in Buffer (LPIB):** Indicates the number of bytes that have been received off the link. This register will count from 0 to the value in the Cyclic Buffer Length register (inclusive) for the first round, and then repeat counting from 1 to the value in the Cyclic Buffer Length register again for subsequent rounds. Note that SW may initialize the value for resume operation usage before RUN bit is set for the first time (and if RSM = 1). Once RUN bit is set, the register will become read only until SRST occurs.

25.45 Input/Output Stream Descriptor x Cyclic Buffer Length (ISD0CBL)—Offset 88h

Register for Input/Output Stream Descriptor x Cyclic Buffer Length.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**Cyclic Buffer Length (CBL):** Indicates the number of bytes in the complete cyclic buffer. CBL must represent an integer number of samples. Link Position in Buffer (LPIB) will be reset when it reaches this value. Software may only write to this register after Global Reset, Controller Reset, or Stream Reset has occurred. This value should only be modified when the RUN bit is '0'. Once the RUN bit has been set to enable the engine, software must not write to this register until after the next reset is asserted, or transfers may be corrupted.

25.46 (ISD0LVI_ISD0FIFOW)—Offset 8Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 40000h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD1_1):** Reserved.
18:16	4h RO/V	**FIFO Watermark (FIFOW):** Indicates the minimum number of bytes accumulated (for input) or free (for output) in the FIFO before the controller will start an eviction (for input) or fetch (fo r output) of data. The PCH HD Audio controller hardwires the FIFO Watermark, either 32B or 64B based on the following. Bit(2:0) Description [*]000-011 Reserved

§ §

26 SRAM Registers

This chapter documents the registers in Bus: 0, Device 13, Function 3.

26.1 DEVICEVENDORID - Device ID and Vendor ID Register (DEVVENDID) — Offset 0h

Device ID and Vendor ID provided by this register uniquely identifies the particular Device

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + 0h	AEC8086 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	AECh RO/V	**DEVICEVENDORID - Device ID Field (DEVICEID):** Device ID identifies the particular PCI device
15:0	8086h RO/V	**DEVICEVENDORID - Vendor ID Field (VENDORID):** Vendor ID is a unique ID provided by the PCI SIG which identifies the manufacturer of the device

26.2 STATUSCOMMAND — Offset 4h

Command register to programme interrupt disable bus master enable and Memory space enable. Status register to read the errors and aborts

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + 4h	100000 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved Field (RESERVED0):** Reserved
29	0h RW/1C/V	**STATUSCOMMAND- RMA Field (RMA):** Received Master Abort
28	0h RW/1C/V	**STATUSCOMMAND- RTA Field (RTA):** Received Target Abort
27:21	0h RO	**Reserved Field (RESERVED1):** Reserved
20	1h RO	**STATUSCOMMAND- Cap List Field (CAPLIST):** Capabilities List: Indicates that the controller contains a capabilities pointer list
19	0h RO	**STATUSCOMMAND- Intrerrupt Status Field (INTR_STATUS):** Interrupt Status: This bit reflects state of interrupt in the device

Bit Range	Default & Access	Field Name (ID): Description
18:16	0h RO	**Reserved Field (RESERVED2):** Reserved
15:11	0h RO	**Reserved Field (RESERVED3):** Reserved
10	0h RW	**STATUSCOMMAND- Interrupt Disable Field (INTR_DISABLE):** Interrupt Disable
9	0h RO	**Reserved Field (RESERVED4):** Reserved
8	0h RW	**STATUSCOMMAND- SERR Enable Field (SERR_ENABLE):** SERR Enable Not implemented
7:3	0h RO	**Reserved Field (RESERVED5):** Reserved
2	0h RW	**STATUSCOMMAND- BME Field (BME):** Bus Master Enable
1	0h RW	**STATUSCOMMAND- MSE Field (MSE):** Memory Space Enable
0	0h RW/V	**STATUSCOMMAND. IOSR FIELD (IOSE):** Controls a device's response to I/O Space accesses. A value of 0 disables the device response. A value of 1 allows the device to respond to I/O Space accesses. State after RST# is 0. NOTE: This bit does not exist in the PMC IOSF2OCP bridge. It is shadowed in the PSF3 fabric. Using /V in AccessType so PMC cluster validation does not assume this bit will read back what was written.

26.3 REVCLASSCODE — Offset 8h

Revision ID register identifies revision of particular device and Class Code register is used to identify generic function of the device

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + 8h	5000000 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	50000h RO/V	**REVCLASSCODE - Class code Field (CLASS_CODES):** Class Code register is read-only and is used to identify the generic function of the device and in some cases a specific register-level programming interface
7:0	0h RO/V	**REVCLASSCODE - Revision ID Field (RID):** Revision ID identifies the revision of particular PCI device.

26.4 CLLATHEADERBIST — Offset Ch

Cache Line size as RW with def 0 Latency timer RW with def 0 Header type with Type 0 configuration header and Reserved BIST register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved Field (RESERVED0):** Reserved
23	0h RO	**CLLATHEADERBIST - MultiFunction Device Field (MULFNDEV):** Multi-Function Device
22:16	0h RO	**CLLATHEADERBIST - Header Type Field (HEADERTYPE):** Header Type: Implements Type 0 Configuration header
15:8	0h RO	**CLLATHEADERBIST - Latency Timer Field (LATTIMER):** Latency Timer:. This register is implemented as R/W with default as 0
7:0	0h RW	**CLLATHEADERBIST - Cache Line Size Field (CACHELINE_SIZE):** Cacheline Size

26.5 BAR — Offset 10h

Base Address Register low [31:2] type[2:1] in 32bit or 64bit addr range and memory space indicator [0]

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + 10h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**BAR -Base Address Field (BASEADDR):** Base Address Register Low Base address of the OCP fabric memory space. Taken from Strap values as ones
11:4	0h RO	**BAR -Size Field (SIZEINDICATOR):** Size Indicator RO Always returns 0 The size of this register depends on the size of the memory space
3	0h RO	**BAR -Prefetchable Field (PREFETCHABLE):** Prefetchable: Indicates that this BAR is not prefetchable
2:1	0h RO	**BAR -Type Field (TYPE):** If BAR_64b_EN is 0 then 00 indicates BAR lies in 32bit address range If BAR_64b_EN is 1 then 10 Indicates BAR lies in 64 bit address range
0	0h RO	**BAR - Message Space Field (MESSAGE_SPACE):** Memory Space Indicator: 0 indicates this BAR is present in the memory space.

26.6 BAR -Base Address Register High (BAR_HIGH) — Offset 14h

Base Address Register High enabled if [2:1] of BAR_type_LOW is 10

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + 14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**BAR -Base Address High Field (BASEADDR_HIGH):** Base Address high - MSB

26.7 BAR1 — Offset 18h

Base Address Register1 accesses to PCI configuration space and is always 4K type in [2:1] and memory space indicator in [0]

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + 18h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**BAR1 - Base Address Field (BASEADDR1):** Base Address1 This field is present if BAR1 is enabled through private configuration space.
11:4	0h RO	**BAR1 -Size Field (SIZEINDICATOR1):** Always is 0 as minimum size is 4K
3	0h RO	**BAR1 - Prefetchable Field (PREFETCHABLE1):** Prefetchable: Indicates that this BAR is not prefetchable.
2:1	0h RO	**BAR1 -Type Field (TYPE1):** If BAR_64b_EN is 0 then 00 indicates BAR lies in 32bit address range If BAR_64b_EN is 1 then 10 Indicates BAR lies in 64 bit address range
0	0h RO	**BAR1 - Message Space Field (MESSAGE_SPACE1):** Memory Space Indicator: 0 Indicates this BAR is present in the memory space

26.8 BAR1 -Base Address Register1 High (BAR1_HIGH) — Offset 1Ch

Base Address Register1 High enabled only if [2:1] of BAR1 register is 10

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + 1Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**BAR1 -Base Address High Field (BASEADDR1_HIGH):** Base Address: Base address of the OCP fabric memory space. Taken from Strap values as ones

26.9 BAR2 — Offset 20h

BAR -Base Address Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + 20h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**BASEADDR:**
1	0h RO	**RESERVED0:** Reserved
0	0h RO	**MESSAGE_SPACE:**

26.10 SUBSYSTEMID — Offset 2Ch

SVID register along with SID register is to distinguish subsystem from another

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**SUBSYSTEMID:** Subsystem ID: This register is implemented for any function that can be instantiated more than once in a given system.
15:0	0h RW/O	**SUBSYSTEMID -Subsystem Vendor Field (SUBSYSTEMVENDORID):** Subsystem Vendor ID: This register must be implemented for any function that can be instantiated more than once in a given system

26.11 EXPANSION ROM base address (EXPANSION_ROM_BASEADDR) — Offset 30h

EXPANSION ROM base address register is a RO indicates support for expansion ROMs

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + 30h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**EXPANSION ROM base address field (EXPANSION_ROM_BASE):** Expansion ROM Base Address: Value of all zeros indicates no support for Expansion ROM

26.12 CAPABILITYPTR — Offset 34h

Capabilities Pointer register indicates what the next capability is

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + 34h	80 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved Field (RESERVED0):** Reserved
7:0	80h RO	**CAPABILITYPTR - Capabilities Pointer Field (CAPPTR_POWER):** Capabilities Pointer: Indicates what the next capability is.

26.13 INTERRUPTREG — Offset 3Ch

Interrupt line Register isn't used in Bridge directly Interrupt Pin register reflects the IPIN value in private config space. Min_gnt register indicating the req of latency timers and max_lat register max latenc

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + 3Ch	100 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**INTERRUPTREG - Max Latency Field (MAX_LAT):** Value of 0 indicates device has no major requirements for the settings of latency timers
23:16	0h RO	**INTERRUPTREG - Min Gnt Field (MIN_GNT):** Value of 0 indicates device has no major requirements for the settings of latency timers // Re-Check Desc Padma
15:12	0h RO	**Reserved Field (RESERVED0):** Reserved
11:8	1h RO	**INTERRUPTREG - Interrupt Pin Field (INTPIN):** Interrupt Pin: Value in this register is reflected from the IPIN value in the private configuration space.
7:0	0h RW	**INTERRUPTREG - Int Line Field (INTLINE):** Interrupt Line: It is used to communicate to software the interrupt line to which the interrupt pin is connected

26.14 POWERCAPID — Offset 80h

PowerManagement Capability ID register points to next capability structure and power mgmnt capability with Power management capabilities register for PME support and version

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + 80h	30001 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	0h RO	**POWERCAPID - PME Support Field (PMESUPPORT):** This 5-bit field indicates the power states in which the function can assert the PME#
26:19	0h RO	**Reserved Field (RESERVED0):** Reserved
18:16	3h RO	**POWERCAPID - Version Field (VERSION):** Indicates support for Revision 1.2 of the PCI Power Management Specification
15:8	0h RO	**POWERCAPID - Next Cap Field (NXTCAP):** Next Capability: Points to the next capability structure.
7:0	1h RO	**POWERCAPID - Power Capability ID Field (POWER_CAP):** Power Management Capability: Indicates this is power management capability

26.15 PMECTRLSTATUS — Offset 84h

power management control and status register to set and read PME status PME enable No Soft reset and power state

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + 84h	8 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved Field (RESERVED0):** Reserved
15	0h RW/1C/V	**PMECTRLSTATUS - PME Status Field (PMESTATUS):** PME Status
14:9	0h RO	**Reserved Field (RESERVED1):** Reserved
8	0h RW	**PMECTRLSTATUS - PME Enable Field (PMEENABLE):** PME Enable
7:4	0h RO	**Reserved Field (RESERVED2):** Reserved
3	1h RO	**PMECTRLSTATUS - No Soft Reset Field (NO_SOFT_RESET):** This bit indicates that devices transitioning from D3hot to D0 because of Powerstate commands do not perform an internal reset
2	0h RO	**Reserved Field (RESERVED3):** Reserved
1:0	0h RW	**PMECTRLSTATUS - Power State Field (POWERSTATE):** Power State: This field is used both to determine the current power state and to set a new power state

26.16 PCI DEVICE IDLE CAPABILITY RECORD (PCIDEVIDLE_CAP_RECORD) — Offset 90h

PCI Device Vendor Specific Capability register defines Vendor specific Capability ID, revision , length , next capability and CAPID

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + 90h	F0140009 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	Fh RO	**PCIDEVIDLE_CAP_REG - Vendor Cap Field (VEND_CAP):** Vendor Specific Capability ID
27:24	0h RO	**PCIDEVIDLE_CAP_REG - Revision IDField (REVID):** Revision ID of capability structure
23:16	14h RO	**PCIDEVIDLE_CAP_REG - Cap Length Field (CAP_LENGTH):** Vendor Specific Capability Length
15:8	0h RO	**PCIDEVIDLE_CAP_REG - Next Capability Field (NEXT_CAP):** Next Capability
7:0	9h RO	**PCIDEVIDLE_CAP_REG - Capability ID Field (CAPID):** Capability ID

26.17 DEVID VENDOR SPECIFIC REG (DEVID_VEND_SPECIFIC_REG) — Offset 94h

Extended Vendor capability register for VSEC Length revision and ID

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + 94h	1400010 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	14h RO	**DEVID VENDOR SPECIFIC REG - Vendor Specific ID Field (VSEC_LENGTH):** Vendor Specific Extended Capability Length
19:16	0h RO	**DEVID VENDOR SPECIFIC REG - Vendor Specific Revision Field (VSEC_REV):** Vendor specific Extended Capability revision
15:0	10h RO	**DEVID VENDOR SPECIFIC REG - Vendor Specific Length Field (VSECID):** Vendor Specific Extended Capability ID

26.18 D0I3_CONTROL_SW_LTR_MMIO_REG — Offset 98h

Software location pointer in MMIO space as an offset specified by BAR

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + 98h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**D0I3_CONTROL_SW_LTR_MMIO_REG - SW LTR Dword Offset Field (SW_LAT_DWORD_OFFSET):** SW LTR Update MMIO Offset Location (SWLTRLOC)
3:1	0h RO	**D0I3_CONTROL_SW_LTR_MMIO_REG - SW LTR BAR Number Field (SW_LAT_BAR_NUM):** Indicates that the SW LTR update MMIO location is always at BAR0
0	0h RO	**D0I3_CONTROL_SW_LTR_MMIO_REG - SW LTR Valid Field (SW_LAT_VALID):** This value is reflected from the SW LTR valid strap at the top level

26.19 DEVICE_IDLE_POINTER_REG — Offset 9Ch

Device IDLE pointer register giving details on Device MMIO offset location BAR NUM and D0i3 Valid Strap

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + 9Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**DEVICE_IDLE_POINTER_REG - D0i3 Dword Offset Field (DWORD_OFFSET):** contains the location pointer to the SW LTR register in MMIO space as an offset from the specified BAR
3:1	0h RO	**DEVICE_IDLE_POINTER_REG - BAR NUM Field (BAR_NUM):** Bar num: Indicates that the D0i3 MMIO location is always at BAR0
0	0h RO	**DEVICE_IDLE_POINTER_REG - D0i3 Valid Field (VALID):** Valid: This value is reflected from the D0i3 valid strap at the top level.

26.20 D0I3_MAX_POW_LAT_PG_CONFIG — Offset A0h

D0idle_Max_Power_On_Latency register set at boot and Power control enable register to enable communication with the PGCB block below the Bridge

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + A0h	800 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved Field (RESERVED0):** Reserved
21	0h RW	**D0I3_MAX_POW_LAT_PG_CONFIG - HAE Field (HAE):** Hardware Autonomous Enable
20	0h RO	**Reserved Field (RESERVED1):** Reserved
19	0h RW	**D0I3_MAX_POW_LAT_PG_CONFIG - Sleep Enable Field (SLEEP_EN):** Sleep Enable

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW	**D0I3_MAX_POW_LAT_PG_CONFIG - D3 Hen Field (PGE):** DEVIDLE Enable (DEVIDLEN): If ?1? then the function will power gate when idle and the DevIdle register (DevIdleC[2] = ?1?) is set.
17	0h RW	**D0I3_MAX_POW_LAT_PG_CONFIG - Device Idle En Field (I3_ENABLE):** D3-Hot Enable (D3HEN): If ?1?, then function will power gate when idle and the PMCSR[1:0] register in the function =?11? (D3).
16	0h RW	**D0I3_MAX_POW_LAT_PG_CONFIG - PMC Request Enable Field (PMCRE):** PMCRE: PMC Request Enable
15:13	0h RO	**Reserved Field (RESERVED2):** Reserved
12:10	2h RW/O	**D0I3_MAX_POW_LAT_PG_CONFIG - Power Latency Scale Field (POW_LAT_SCALE):** Power On Latency Scale
9:0	0h RW/O	**D0I3_MAX_POW_LAT_PG_CONFIG - Power Latency Value Field (POW_LAT_VALUE):** Power On Latency value

26.21 GEN_REGRW1 - General Purpose Read Write Register1 (GEN_REGRW1) — Offset B0h

General Purpose PCI Read Write Register1

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + B0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REGRW1 - General Purpose Read Write Field (GEN_REG_RW1):** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg1 Out of Band signal

26.22 GEN_REGRW2 — Offset B4h

General Purpose PCI Read Write Register2

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + B4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REGRW2 - General Purpose Read Write Field (GEN_REG_RW2):** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg2 Out of Band signal

26.23 GEN_REGRW3 — Offset B8h

General Purpose PCI Read Write Register3

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + B8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REGRW3 - General Purpose Read Write Field (GEN_REG_RW3):** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg3 Out of Band signal

26.24 GEN_REGRW4 — Offset BCh

General Purpose PCI Read Write Register4

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + BCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GEN_REGRW4 - General Purpose Read Write Field (GEN_REG_RW4):** General Purpose PCI Register: This register value is brought out as oob_gen_pci_reg4 Out of Band signal

26.25 GEN_INPUT_REG — Offset C0h

General Purpose Input Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + C0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**GEN_INPUT_REG - General Purpose Input Field (GEN_REG_INPUT_RW):** General Purpose Input Register: This register value reflects the value of oob_gen_input_pci Out of Band signal

26.26 Manufacturers ID (MANID) — Offset F8h

MAN ID Register

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:3] + F8h	4000F1C h

Bit Range	Default & Access	Field Name (ID): Description
31:0	4000F1Ch RO/V	**Manufacturers ID - MAN ID (MANID):** Manufacturer ID: Default value comes from straps.

§ §

27 SPI Controller Registers

This chapter documents the registers in Bus: 0, Device 13, Function 2.

27.1 Device ID and Vendor ID (BIOS_SPI_DID_VID) — Offset 0h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 0h	31968086 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	3196h RO/V	**Device Identification (DID):** Identifier for the SPI Flash Controller in Host Root Space. The upper 9-bits of this field can be overridden by the SetIDValue IOSF-SB Message. The value of the lower 7 bits is parameterized to allow different SOCs to configure the SPI IP. (The recommendation is to use pin-ties for the parameter to allow IP reuse without re-compilation.)
15:0	8086h RO	**Vendor Identification (VID):** This field identifies the manufacturer of the device. Valid vendor identifiers are allocated by the PCI SIG to ensure uniqueness. 0x8086 indicates Intel

27.2 Status and Command (BIOS_SPI_STS_CMD) — Offset 4h

This is a standard PCI config register. See the PCI spec for bit descriptions.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 4h	400 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C/V	**Detected Parity Error (DPE):** See the PCI spec.
30	0h RW/1C/V	**Signaled System Error (SSE):** See the PCI spec.
29	0h RO	**Received Master Abort (RMA):** See the PCI spec.
28	0h RO	**Received Target Abort (RTA):** See the PCI spec.
27	0h RW/1C/V	**Signaled Target Abort (STA):** See the PCI spec.
26:25	0h RO	**Devsel Timing (DEVT):** See the PCI spec.
24	0h RO	**Master Data Parity Error (MDPE):** See the PCI spec.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RO	**Fast Back to Back Capable (FBTBC):** Has no meaning on the internal backbone.
22	0h RO	**Reserved (RSVD0):** Reserved
21	0h RO	**66 Mhz Capable (MCAP):** Not 66 MHz capable device. Has no meaning on the internal backbone.
20	0h RO	**Capablities List (CAPL):** See the PCI spec.
19	0h RO	**Interrupt Status (INTS):** See the PCI spec.
18:11	0h RO	**Reserved (RSVD1):** Reserved
10	1h RO	**Interrupt Disable (INTD):** See the PCI spec.
9	0h RO	**Fast Back to Back Enable (FBTBEN):** See the PCI spec.
8	0h RW	**System Error Enable (SERREN):** See the PCI spec.
7	0h RO	**Reserved (RSVD):** Reserved
6	0h RW	**Parity Error Response (PERRR):** See the PCI spec.
5	0h RO	**VGA Palette Snoop (VGAPS):** See the PCI spec.
4	0h RO	**Memory Write and Invalidate Enable (MWRIEN):** See the PCI spec.
3	0h RO	**Special Cycles (SPCYC):** See the PCI spec.
2	0h RW	**Bus Master Enable (BME):** See the PCI spec.
1	0h RW	**Memory Space Enable (MSE):** Memory Space Enable may default False because the BIOS boot fetch decoding is hardcoded and does not rely on a BAR in config space.
0	0h RO	**IO Space Enable (IOSE):** See the PCI spec.

27.3 Revision ID and Class Code (BIOS_SPI_CC_RID) — Offset 8h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 8h	C800000 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	Ch RO	**Base Class Code (BCC):**

Bit Range	Default & Access	Field Name (ID): Description
23:16	80h RO	**Sub-Class Code (SCC):**
15:8	0h RO	**Programming Interface (PI):**
7:0	0h RO/V	**Revision ID (RID):** Indicates the part revision. This will reset to 0 but will overridden by the SetID IOSF-SB message during the power-up sequence.

27.4 BIST, Header Type, Latency Timer, Cache Line Size (BIOS_SPI_BIST_HTYPE_LT_CLS) — Offset Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23	0h RO	**Multi-function Device (MFD):** See the PCI spec.
22:16	0h RO	**Header Type (HTYPE):** See the PCI spec.
15:8	0h RO	**Latency Timer (LT):** See the PCI spec.
7:0	0h RO	**Cacheline Size (CLSZ):** See the PCI spec.

27.5 SPI BAR0 MMIO (BIOS_SPI_BAR0) — Offset 10h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 10h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**Memory BAR (MEMBAR):** Software programs this register with the base address of the device's memory region. The Host/BIOS MMIO registers in the flash controller are offset from this BAR.
11:4	0h RO	**Memory Size (MEMSIZE):** Hardwired to 0 to indicate 4KB of memory space.
3	0h RO	**PREFETCH:** A device can mark a range as prefetchable if there are no side effects on reads, the device returns all bytes on reads regardless of the byte enables.
2:1	0h RO	**TYP:** Hardwired to 0 to indicate that Base register is 32 bits wide and mapping can be done anywhere in the 32-bit Memory Space.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**Memory Space Indicator (MEMSPACE):** Hardwired to 0 to identify a Memory BAR.

27.6 CSXE BAR1 Direct Reads (CSXE_SPI_BAR1) — Offset 14h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RW	**Memory BAR (MEMBAR):** Software programs this register with the base address of the device's memory region. CSME direct reads to flash region 2.
22:4	0h RO	**Memory Size (MEMSIZE):** Hard wired to 0 to indicate 8MB of memory space.
3	0h RO	**PREFETCH:** A device can mark a range as prefetchable if there are no side effects on reads, the device returns all bytes on reads regardless of the byte enables.
2:1	0h RO	**TYP:** Hardwired to 0 to indicate that Base register is 32 bits wide and mapping can be done anywhere in the 32-bit Memory Space.
0	0h RO	**Memory Space Indicator (MEMSPACE):** Hardwired to 0 to identify a Memory BAR.

27.7 CSXE BAR2 Huffman Decompression (CSXE_SPI_BAR2) — Offset 18h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 18h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**Memory BAR (MEMBAR):** Software programs this register with the base address of the device's Huffman decompression region. The concatenation of this field and MEMSIZE is referred to as CSME Hardware Decompression Flash Range Base (MEHDFRB) within this document. Software programs this register with the base address of the Huffman decompression region. **The base address must be 32MB aligned.** The size is fixed at 16MB.
23:4	0h RO	**Memory Size (MEMSIZE):** Hard wired to 0 to indicate 16MB of memory space.
3	0h RO	**PREFETCH:** A device can mark a range as prefetchable if there are no side effects on reads, the device returns all bytes on reads regardless of the byte enables.
2:1	0h RO	**TYP:** Hardwired to 0 to indicate that Base register is 32 bits wide and mapping can be done anywhere in the 32-bit Memory Space.
0	0h RO	**Memory Space Indicator (MEMSPACE):** Hardwired to 0 to identify a Memory BAR.

27.8 CSXE BAR3 Touch BAR (CSXE_SPI_BAR3) — Offset 1Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 1Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RW	**Memory BAR (MEMBAR):** Software programs this register with the base address of the Touch region. The base address must be 128 kB aligned. The size is fixed at 128 kBytes.
16:4	0h RO	**Memory Size (MEMSIZE):** Hard wired to 0 to indicate 128KB of memory space.
3	0h RO	**PREFETCH:** A device can mark a range as prefetchable if there are no side effects on reads, the device returns all bytes on reads regardless of the byte enables.
2:1	0h RO	**TYP:** Hardwired to 0 to indicate that Base register is 32 bits wide and mapping can be done anywhere in the 32-bit Memory Space.
0	0h RO	**Memory Space Indicator (MEMSPACE):** Hardwired to 0 to identify a Memory BAR.

27.9 CSXE BAR4 BIOS Direct Reads (CSXE_SPI_BAR4) — Offset 20h

This BAR enables the CSE firmware to efficiently read and authenticate the BIOS image in SPI flash without changing the SPI controller's fundamental requirement that regions be non-overlapping. The SPI controller supports two independent regions for BIOS code, region 1 and region 6. The architecture defines CSE access to either region, but the implementation only requires region 1. The CSE direct read of the BIOS region is directed to either region 1 or region 6 based on the CSE's PCI Config register CSXEFCTRL.BRS bit. Address translation from the BAR to the flash linear address is similar to other CSE direct reads, meaning address BAR+n maps to region_base + n. Range checking is also similar. The direct read will be UR'd if the address exceeds the MIN(16MB, ars), where ars is the size of the active BIOS region.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 20h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**Memory BAR (MEMBAR):** Software programs this register with the base address of the device's memory region. CSME direct reads to the active BIOS region in flash. The size is fixed at 16MB.
23:4	0h RO	**Memory Size (MEMSIZE):** Hard wired to 0 to indicate 16MB of memory space
3	0h RO	**PREFETCH:** A device can mark a range as prefetchable if there are no side effects on reads, the device returns all bytes on reads regardless of the byte enables.
2:1	0h RO	**TYP:** Hardwired to 0 to indicate that Base register is 32 bits wide and mapping can be done anywhere in the 32-bit Memory Space.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**Memory Space Indicator (MEMSPACE):** Hardwired to 0 to identify a Memory BAR.

27.10 Cardbus CIS Pointer (CSXE_SPI_CCP) — Offset 28h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 28h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD):** Reserved

27.11 Subsystem and Vendor ID (BIOS_SPI_SID_SVID) — Offset 2Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 2Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW/O	**Subsystem ID (SSID):**
15:0	0h RW/O	**Subsystem Vendor ID (SSVID):**

27.12 Expansion ROM Base Address (CSXE_SPI_XRBAR) — Offset 30h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 30h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD):** Reserved

27.13 Capabilities List Pointer (CSXE_SPI_CAPP) — Offset 34h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 34h	44 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD):** Reserved
7:0	44h RO	**Capabilities Pointer (CAPP):** Indicates the pointer for the first entry in the capabilities list

27.14 Maximum Latency, Minimum Grant, Interrupt Pin and Interrupt Line (CSXE_SPI_MAXL_MING_INTP_INTL) — Offset 3Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 3Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD):** Reserved

27.15 CSXE Capabilities List Pointer (CSXE_SPI_CCAPP) — Offset 40h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 40h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD):** Reserved
7:0	0h RO	**CSXE Capabilities Pointer (CAPP):** Indicates the pointer for the first entry in the CSXE capabilities list

27.16 MSI Message Control, Next Pointer and Capability ID (CSXE_SPI_MSIMC_MSINP_MSICID) — Offset 44h

These PCI Config registers does not support MSI functionality, therefore the SSIE and MSIE interrupts must not be enabled by software.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 44h	5005 h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD):** Reserved
24	0h RO	**Per Vector Masking Capable (PVMC):** This function does not support MSI per vector masking
23	0h RO	**64 bit address capable (XAC):** This function is not capable of sending 64 bit message address
22:20	0h RO	**Multiple Message Enable (MMEN):** Encoded number of interrupt vectors allocated by SW. Value of zero indicates one vector.
19:17	0h RO	**Multiple Message Capable (MMC):** Encoded number of interrupt vectors supported. Value of zero indicates one vector.
16	0h RW	**MSI Enable (MSIE):** If set to '1' CXME MSI interrupt delivery is enabled. When this bit is cleared, prior to returning the configuration write completion, the device must send any pending MSI(s).
15:8	50h RO	**Next Item Pointer (NXTP):** Indicates the pointer for the next entry in the capabilities list
7:0	5h RO	**Capability ID (CAPID):** Indicates the linked list item as being the MSI Capability registers

27.17 MSI Message Address (CSXE_SPI_MSIMA) — Offset 48h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 48h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**Message Address (MADDR):** DW aligned MSI message address.
1:0	0h RO	**Reserved (RSVD):** Reserved

27.18 MSI Message Data (CSXE_SPI_MSIMD) — Offset 4Ch

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 4Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:0	0h RW	**Message Data (MDAT):** MSI Message Data

27.19 PCI Power Management Capability (CSXE_SPI_PMCAP_PMNP_PMCID) — Offset 50h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 50h	30001 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	0h RO	**PME Support (PMES):** This 5-bit field indicates the power states in which the function may assert PME#. A value of 0b for any bit indicates that the function is not capable of asserting the PME# signal while in that power state. bit(27) X XXX1b - PME# can be asserted from D0 bit(28) X XX1Xb - PME# can be asserted from D1 bit(29) X X1XXb - PME# can be asserted from D2 bit(30) X 1XXXb - PME# can be asserted from D3hot bit(31) 1 XXXXb - PME# can be asserted from D3cold
26	0h RO	**D2 Support (D2S):** This device does not support D2
25	0h RO	**D1 Support (D1S):** This device does not support D1
24:22	0h RO	**Aux Current (AUXC):** Not Applicable
21	0h RO	**Device Specific Initialization (DSI):** See PCI Power Management Interface specification.
20	0h RO	**Reserved (RSVD):** Reserved
19	0h RO	**PME Clock (PMECLK):** Not Applicable
18:16	3h RO	**VER:** Value of 011b indicates that this function complies with rev 1.2 of PCI Power Management Interface Spec
15:8	0h RO	**Next Item Pointer (NXTP):** Indicates the pointer for the next entry in the capabilities list
7:0	1h RO	**Capability ID (CAPP):** Indicates the linked list item as being the PCI Power Management registers

27.20 PCI Power Management Control and Status (CSXE_SPI_PMD_PMCSRBSE_PMCSR) — Offset 54h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 54h	8 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15	0h RO	**PME Status (PMESTS):** See PCI Power Management Interface specification.
14:13	0h RO	**Data Scale (DS):** Not Applicable
12:9	0h RO	**Data Select (DSEL):** Not Applicable
8	0h RO	**PME Enable (PMEEN):** See PCI Power Management Interface specification.
7:4	0h RO	**Reserved (RSVD1):** Reserved
3	1h RO	**No Soft Reset (NSR):** When set to 1, this bit indicates that devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset. Configuration Context is preserved. Upon transition from the D3hot to the D0 Initialized state, no additional operating system intervention is required to preserve Configuration Context beyond writing the PowerState bits.
2	0h RO	**Reserved (RSVD2):** Reserved
1:0	0h RO	**Power State (PWRST):** See PCI Power Management Interface specification.

27.21 CSXE Flash Control (CSXE_SPI_CSXEFCTRL) — Offset 80h

Note: The BIOS Region Select bit is reset on early_boot_side_rst_b, not on the CSXE partition reset.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 80h	3 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	0h RO	**Reserved (RSVD):** Reserved
26:6	0h RW	**CSXE Hardware Decompression LUT Base (HDLUTB):** Specifies bits [26:6] of the base of the Look-Up-Table as an offset from the CSME Flash Region 2 Base. See 'CSME Region HDLUT' for a description of how this field is used.
5	0h RW/L	**CSXE HD Page Size (HDPAGESIZE):** Defines the page size within the hardware decompression range. 0 : 1KB (default) 1 : 4KB This register is locked by the CSME Flash Address Lockdown MEFAL register.
4	0h RW/L	**CSXE Hardware Decompression Enable (HDEN):** Set to '1' by CSME FW after programming the CSME HE Flash Range Base and limit registers to enable HW Decompression. This register is locked by the CSME Flash Address Lockdown MEFAL register.
3	0h RW/P/L	**BIOS Region Select (BRS):** When set to a '1' by CSME FW, BIOS direct reads to SPI flash are mapped to the Secondary BIOS flash region 6. When this bit is '0' (default), BIOS direct reads to SPI flash are mapped to flash region 1. Note: The Flash controller only allows this bit to be written if the Server Feature Disable Fuse 6 is '0'. This bit is reset on early_boot_side_rst_b.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW/L	**CSXE Flash Address Lockdown (FAL):** When set, this bit locks fields in CSXEFCTRL and CSMEHDFL. This register locks itself.
1	1h RW	**CSXE Direct Read Prefetch Enable (DRPE):** When set to a '1', CSME direct reads are prefetched in the Flash Controller. When this bit is set to '1' the Cache Enable (MEDRCE) must also be set to '1'.
0	1h RW	**CSXE Direct Read Cache Enable (DRCE):** When set to a '1', CSME direct reads are cached in the Flash Controller. Note that if CSME direct read caching is disabled while data has already been cached internally, subsequent CSME direct reads will continue to return data from the cache until the cache is invalidated.

27.22 CSXE Hardware Decompression Flash Limit (CSXE_SPI_CSXEHDFL) — Offset 84h

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + 84h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW/L	**CSXE HD Flash Range Limit (HDFRL):** CSME Hardware Decompression Flash Range Limit. The limit is 4KB aligned. This register is locked by the CSME Flash Address Lockdown MEFAL register.
11:0	0h RO	**Reserved (RSVD):** Reserved

27.23 SPI Unsupported Request Status (BIOS_SPI_UR_STS_CTL) — Offset D0h

See the IOSF Specification for required behavior.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + D0h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Reserved (RSVD):** Reserved
1	0h RW/1C/V	**Unsupported Request Detected (URD):** Set to 1 by hardware upon detecting an Unspported Request that is not considered an Advisory Non-Fatal error. Cleared to 0 when software writes a 1 to this register.
0	0h RW	**Unsupported Request Reporting Enabled (URRE):** If set to 1 by software, the flash controller generates a DoSERR sideband message when the URD bit transitions from 0 to 1.

27.24 B-Unit Copy of the BIOS Decode Enable register for SPI (B_CR_BDE_SPI) — Offset D8h

Affects B-Unit SAD for BIOS regions, only when BIOS is resident in SPI.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + D8h	FFCF h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RESERVED_0):** Reserved
15	1h RO	**F8-FF Enable (EF8):** Enables decoding of 512K of the following BIOS ranges: FFF80000h-FFFFFFFFh and FFB80000h-FFBFFFFFh.
14	1h RW/V	**F0-F8 Enable (EF0):** Enables decoding of 512K of the following BIOS ranges: FFF00000h-FFF7FFFFh and FFB00000h-FFB7FFFFh.
13	1h RW/V	**E8-EF Enable (EE8):** Enables decoding of 512K of the following BIOS ranges: FFE80000h-FFEFFFFFh and FFA80000h-FFAFFFFFh.
12	1h RW/V	**E0-E8 Enable (EE0):** Enables decoding of 512K of the following BIOS ranges: FFE00000h-FFE7FFFFh and FFA00000h-FFA7FFFFh.
11	1h RW/V	**D8-DF Enable (ED8):** Enables decoding of 512K of the following BIOS ranges: FFD80000h-FFDFFFFFh and FF980000h-FF9FFFFFh.
10	1h RW/V	**D0-D7 Enable (ED0):** Enables decoding of 512K of the following BIOS ranges: FFD00000h-FFD7FFFFh and FF900000h-FF97FFFFh.
9	1h RW/V	**C8-CF Enable (EC8):** Enables decoding of 512K of the following BIOS ranges: FFC80000h-FFCFFFFFh and FF880000h-FF8FFFFFh.
8	1h RW/V	**C0-C7 Enable (EC0):** Enables decoding of 512K of the following BIOS ranges: FFC00000h-FFC7FFFFh and FF800000h-FF87FFFFh.
7	1h RW/V	**Legacy F Segment Enable (LFE):** This enables the decoding of the legacy 64KB range at F0000h-FFFFFh.
6	1h RW/V	**Legacy E Segment Enable (LEE):** This enables the decoding of the legacy 64KB range at E0000h-EFFFFh.
5:4	0h RO	**Reserved (RESERVED_1):** Reserved
3	1h RW/V	**70-7F Enable (E70):** Enables decoding of 1MB of the following BIOS ranges: FF700000h-FF7FFFFFh and FF300000h-FF3FFFFFh.
2	1h RW/V	**60-6F Enable (E60):** Enables decoding of 1MB of the following BIOS ranges: FF600000h-FF6FFFFFh and FF200000h-FF2FFFFFh.
1	1h RW/V	**50-5F Enable (E50):** Enables decoding of 1MB of the following BIOS ranges: FF500000h-FF5FFFFFh and FF100000h-FF1FFFFFh.
0	1h RW/V	**40-4F Enable (E40):** Enables decoding of 1MB of the following BIOS ranges: FF400000h-FF4FFFFFh and FF000000h-FF0FFFFFh.

27.25 BIOS Control (BIOS_SPI_BC) — Offset DCh

This register collects bits that were previously distributed in various IPs and input to the SPI controller via discrete wires.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + DCh	28 h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD):** Reserved
11	0h RW/L	**Async SMI Enable for BIOS Write Protection (ASE_BWP):** When set to '1', the flash controller will generate an SMI when it blocks a BIOS write or erase. The value in this field can be written by software as long as the BIOS Interface Lock-Down (BILD) is not set.
10	0h RO/V	**Asynchronous SMI Status (SPI_ASYNC_SS):** Status indication that the SPI Flash Controller has asserted an asynchronous SMI. Hardware clears the bit when it sends the De-assert SMI message. 0: default state 1: SPI flash controller asserted Asynchronous SMI
9	0h RW/L	**OS Function Hide (OSFH):** This bit controls read access over IOSF Primary to SPI's Device ID, Vendor ID PCI Config register. This bit does not affect access to any other PCI Config registers. This bit is locked with BILD. Trusted BIOS must set this bit prior to starting the OS. 0 : DeviceID, VendorID can be read 1 : reads to Device ID, Vendor ID return a UR **Note:** This bit is still implemented as a RW/L bit, but is no longer used by the hardware to return UR as specified above.
8	0h RW/1C/V	**Synchronous SMI Status (SPI_SYNC_SS):** Status indication that the SPI Flash Controller has asserted a synchronous SMI. Hardware clears the bit when it sends the De-assert Synchronous SMI message. 0: default state 1: SPI flash controller asserted Synchronous SMI
7	0h RW/L	**BIOS Interface Lock-Down (BILD):** When set, prevents BBS and ASE_BWP from being changed. This bit can only be written from 0 to 1 once.
6	0h RW/V/L	**Boot BIOS Strap (BBS):** This field determines the destination of accesses to the BIOS memory range. 0 SPI 1 LPC When SPI is selected, the range that is decoded is further qualified by the BIOS Decode Enable (BDE) register. The initial value of this bit is determined by the BBS pinstrap. The value in this field can be overwritten by software as long as the BIOS Interface Lock-Down (BILD) is not set.
5	1h RW/L	**Enable InSMM.STS (EISS):** When this bit is set, the BIOS region is not writable until the CPU is in SMM mode. SPI receives this indication either via the InSMM.STS bit discrete wire input or via SAI. If this bit [5] is set, then WPD must be a '1' and the CPU must be in SMM mode in order to write to BIOS regions of SPI Flash. If this bit [5] is clear, then SMM mode is a don't care.This bit is locked by LE (bit 1 of this register).
4	0h RO/V	**Top Swap Status (TSS):** This bit provides a read-only path to view the state of the Top Swap bit. It is duplicated here to be consistent with the LPC version of the BC register.
3:2	2h RW	**SPI Read Configuration (SRC):** These bits are located in PCI Config space to allow them to be set early in the boot flow. This 2-bit field controls two policies related to BIOS reads on the SPI interface: Bit 3 - Prefetch Enable Bit 2 - Cache Disable Settings are summarized below: Bits 3:2 -- Description 00 **No prefetching, but caching enabled.** Direct Memory reads load the read buffer cache with 'valid' data, allowing repeated reads to the same range to complete quickly 01 **No prefetching and no caching.** One-to-one correspondence of host BIOS reads to SPI cycles. This value can be used to invalidate the cache. 10 **Prefetching and Caching enabled.** This mode is used for long sequences of short reads to consecutive addresses (i.e. shadowing) 11 **Illegal.** Caching must be enabled when Prefetching is enabled. This eliminates the need for a complex prefetch-flushing mechanism
1	0h RW/L	**Lock Enable (LE):** When set, setting the WPD bit will cause a synchronous SMI. When cleared, setting the WPD bit will not cause SMI. Once set, this bit can only be cleared by a PLTRST#. When this bit is set, EISS - bit [5] of this register is locked down.
0	0h RW	**Write Protect Disable (WPD):** When set, access to the BIOS space is enabled for both read and write cycles to BIOS. When cleared, only read cycles are permitted to the SPI flash. When this bit is written from a '0' to a '1' and the LE bit is also set, a synchronous SMI is generated. This ensures that only SMM code can update BIOS.

27.26 Manufacturer's ID (BIOS_SPI_MANID) — Offset F8h

This register is assigned during the boot flow using the SetID message based on values found in the fuse block. All fields except MID will be reset by hardware to zero and set only by the SetIDValue message.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:13, F:2] + F8h	F00 h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD):** Reserved
27:24	0h RO/V	**DOT:** Indicates the dot process.
23:16	0h RO/V	**Manufacturing Stepping Identifier (MSID):** This field is incremented for each stepping of the part. Note that this field can be used by software to differentiate steppings when the Stepping Revision ID may not change.
15:8	Fh RO/V	**Manufacturing Identifier (MID):** 0Fh = Intel
7:0	0h RO/V	**PROC:** Indicates the process. The dot portion of the process is reflected in bits [27:24]

27.27 SSP Control Register 0 (SSCR0)—Offset 0h

The Enhanced SSP Control 0 registers contain twelve different bit fields that control various functions within the Enhanced SSP. All bits must be set to the preferred value before enabling the Enhanced SSP. Note that Writes to reserved bits must be zeroes, and Read values of these bits is undetermined.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**MOD (MOD):** Mode 0 - Normal SSP Mode 1 - reserved
30	0h RW	**ACS (ACS):** Audio Clock Select 0 - Clock selection is determined by the NCS and ECS bits 1 - reserved
29	0h RW	**LNW_CKBIT_ECO716771 (LNW_CKBIT_ECO716771)**
28:24	0h NA	**RSVD0 (RSVD0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**TIM (TIM):** Transmit FIFO Under Run Interrupt Mask 0 - TUR events will generate an SSP interrupt 1 - TUR events will not generate an SSP interrupt
22	0h RW	**RIM (RIM):** Receive FIFO Over Run Interrupt Mask 0 - ROR events will generate an SSP interrupt 1 - ROR events will not generate an SSP interrupt
21	0h RW	**NCS (NCS):** Network Clock Select 0 - Clock selection is determined by ECS bit 1 - Network clock is used to create the SSP's serial clock (SSPSCLK)
20	0h RW	**EDSS (EDSS):** Extended Data Size Select 0 - A zero is preappended to the DSS value which sets the DSS range from 4-16 bits 1 - A one is pre-appended to the DSS value which sets the DSS range from 17-32 bits
19:8	0h RW	**SCR (SCR):** Serial Clock Rate Value (0 to 4095) used to generate transmission rate of SSP. Serial bit rate = SSP clock/(SCR+1), where SCR is decimal integer.
7	0h RW	**SSE (SSE):** Synchronous Serial Port Enable 0 - SSP operation disabled 1 - SSP operation enabled
6	0h RW	**ECS (ECS):** External Clock Select 0 - On-chip clock used to produce the SSP's serial clock (SSPSCLK) 1 - SSPEXTCLK/GPIO pin is used to create the SSP's SSPCLK
5:4	0h RW	**FRF (FRF):** Frame Format 00 - Motorola Serial Peripheral Interface (SPI) 01 - reserved 10 - reserved 11 - reserved
3:0	0h RW	**DSS (DSS):** Data Size Select With EDSS as MSB, value+1 gives data size. Values 4 to 32 allowed.

27.28 SSP Control Register 1 (SSCR1)—Offset 4h

The Enhanced SSP Control 1 registers contain bit fields that control various SSP functions. Bits must be set to the preferred value before enabling the Enhanced SSP. Note that Writes to reserved bits should be zeroes, and Read value of these bits are undetermined.

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**TTELP (TTELP):** TXD Tristate Enable on Last Phase 0 - TXD line will be tristated on same clock edge as TXD is to be flopped 1 - TXD line will be tristated clock edge after TXD is to be flopped
30	0h RW	**TTE (TTE):** TXD Tristate Enable 0 - TXD line will not be tristated 1 - TXD line will be tristated when no transmitting data

Bit Range	Default & Access	Field Name (ID): Description
29:24	0h NA	**Reserved_29_24 (Res_29_24)**
23	0h RW	**RWOT (RWOT):** Receive With Out Transmit 0 - Transmit/Receive mode 1 - Receive without transmit mode
22	0h RW	**TRAIL (TRAIL):** Trailing Byte 0 - Processor based, trailing bytes are handled by processor 1 - DMA based, trailing bytes are handled by DMA
21	0h RW	**TSRE (TSRE):** Transmit Service Request Enable 0 - DMA Service Request is disabled 1 - DMA Service Request is enabled
20	0h RW	**RSRE (RSRE):** Receive Service Request Enable 0 - DMA Service Request is disabled 1 - DMA Service Request is enabled
19	0h RW	**TINTE (TINTE):** Receiver Time-out Interrupt Enable 0 - Receiver Time-out interrupts are disabled 1 - Receiver Time-out interrupts are enabled
18	0h RW	**PINTE (PINTE):** Peripheral Trailing Byte Interrupts Enable 0 - Peripheral Trailing Byte Interrupts are disabled 1 - Peripheral Trailing Byte Interrupts are enabled
17	0h NA	**RSVD1 (RSVD1):** Reserved
16	0h RW	**IFS (IFS):** Invert Frame Signal 0 - Frame polarity is determined by SSP format and PSP polarity bits 1 - Frame signal will be inverted from the normal SSP frame signal (as defined by the SSP format nad PSP polarity bits)
15	0h RW	**STRF (STRF):** Select FIFO for EFWR (test mode bit) (when EFWR=1) 0 - Transmit FIFO is selected for both writes and reads through the SSP Data Register (SSDR) 1 - Receive FIFO is selected for both writes and reads through the SSP Data Register (SSDR)
14	0h RW	**EFWR (EFWR):** Enable FIFO Write/Read (test mode bit) 0 - FIFO write/read special function is disabled (normal SSP operational mode) 1 - FIFO write/read special function is enabled
13:6	0h NA	**Reserved_13_6 (Res_13_6)**
5	0h RW	**MWDS (MWDS):** Microwire Transmit Data Size 0 - 8-bit command words are transmitted 1 - 16-bit command words are transmitted
4	0h RW	**SPH (SPH):** Motorola SPI SSPSCLK phase setting 0 - SSPSCLK is inactive one sycle at the start of a fram and cycle at the end of a frame 1 - SSPSCLK is inactive cycle at the start of a frame and one cycle at the end of a frame
3	0h RW	**SPO (SPO):** Motorola SPI SSPSCLK polarity setting 0 - The inactive or idle state of SSPSCLK is low 1 - The inactive or idle state of SSPSCLK is high
2	0h RW	**LBM (LBM):** Loop-Back Mode (test mode bit) 0 - Normal serial port operation enabled 1 - Output of transmit serial shifter connected to input of receive serial shifter, internally

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**TIE (TIE):** Transmit FIFO Interrupt Enable 0 - Transmit FIFO level interrupt is disabled 1 - Transmit FIFO level interrupt is enabled
0	0h RW	**RIE (RIE):** Receive FIFO Interrupt Enable 0 - Receive FIFO level interrupt is disabled 1 - Receive FIFO level interrupt is enabled

27.29 SSP Status Register (SSSR)—Offset 8h

The Enhanced SSP Status registers contain bits that signal overrun errors as well as the Transmit and Receive FIFO service requests. Each of these hardware-detected events signals an Interrupt request to the Interrupt controller. The Status register also contains flags that indicate when the Enhanced SSP is actively transmitting data, when the Transmit FIFO is not full, and when the Receive FIFO is not empty. One Interrupt signal is sent to the Interrupt Controller for each SSP. These events can cause an Interrupt: End-of-Chain, Receiver Time-out, Peripheral Trailing Byte, Receive FIFO overrun, Receive FIFO request, and Transmit FIFO request.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: F004h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h NA	**RSVD2 (RSVD2):** Reserved
21	0h RW/1C	**TUR (TUR):** Transmit FIFO Under Run 0 - Transmit FIFO has not experienced an under run 1 - Attempted read from the transmit FIFO when the FIFO was empty, request interrupt
20	0h RW/1C	**EOC (EOC):** End of Chain 0 - DMA has not signaled an end of chain condition 1 - DMA has signaled an end of chain condition
19	0h RW/1C	**TINT (TINT):** Receiver Time-out Interrupt 0 - No receiver time-out pending 1 - Receiver time-out pending
18	0h RW/1C	**PINT (PINT):** Peripheral Trailing Byte Interrupt 0 - No peripheral trailing byte interrupt pending 1 - Peripheral trailing byte interrupt pending
17:16	0h NA	**RSVD5 (RSVD5):** Reserved
15:12	Fh NA	**RSVD6 (RSVD6):** Reserved
11:8	0h NA	**RSVD7 (RSVD7):** Reserved
7	0h RW/1C	**ROR (ROR):** Receive FIFO Overrun 0 - Receive FIFO has not experienced an overrun 1 - Attempted data write to full receive FIFO, request interrupt

Bit Range	Default & Access	Field Name (ID): Description
6	0h RO	**RFS (RFS):** Receive FIFO Service Request 0 - Receive FIFO level is at or below RFT threshold (RFT), or SSP disabled 1 - Receive FIFO level exceeds RFT threshold (RFT), request interrupt
5	0h RO	**TFS (TFS):** Transmit FIFO Service Request 0 - Transmit FIFO level exceeds the TFT threshold (TFT+1), or SSP disabled 1 - Transmit FIFO level is at or below TFT threshold (TFT+1), request interrupt
4	0h RO	**BSY (BSY):** SSP Busy 0 - SSP is idle or disabled 1 - SSP currently transmitting or receiving a frame
3	0h RO	**RNE (RNE):** Receive FIOF Not Empty 0 - Receive FIFO is empty 1 - Receive FIFO is not empty
2	1h RO	**TNF (TNF):** Transmit FIFO Not Full 0 - Transmit FIFO is full 1 - Transmit FIFO is not full
1:0	0h RO	**RSVD8 (RSVD8):** Reserved

27.30 SSP Interrupt Test Register (SSITR)—Offset Ch

The read-write SSP Interrupt Test registers should be used only for testing purposes. Writing a 1 to the test transmit FIFO request SSITR.TTFS, bit 5, will generate a non-maskable Interrupt strobe signal to the Interrupt controller, and a DMA request for the Transmit FIFO. Writing a 1 to the test receive FIFO request SSITR.TRFS, bit 6, will generate a non-maskable Interrupt strobe signal to the Interrupt controller, and a DMA request for the Receive FIFO. Writing a 1 to the test receive FIFO overrun bit SSITR.TROR, bit 7, will generate a non-maskable Interrupt strobe signal to the Interrupt controller only, no DMA request will be made. Setting any of these bits will also cause the corresponding status bit(s) to be set in the Enhanced SSP Status register (SSSR). The Interrupt and/or service request, caused by the setting of one of these test bits, will remain active until the test bit is cleared by writing a 0 it. Note that Writes to reserved bits must be zeroes, and Read value of these bits are undetermined.

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h NA	**RSVD9 (RSVD9):** Reserved
7	0h RW	**TROR (TROR):** Test Receive FIFO overrun 0 - No receive FIFO overrun service request 1 - Generates non-maskable interrupt to CPU. No DMA request is generated
6	0h RW	**TRFS (TRFS):** Test Receive FIFO service request 0 - No receive FIFO service request 1 - Generates non-maskable interrupt to CPU and a DMA request for receive FIFO

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**TTFS (TTFS):** Test Transmit FIFO service request 0 - No transmit FIFO service request pending 1 - Generates non-maskable interrupt to CPU and a DMA request for transmit FIFO
4:0	0h NA	**RSVD10 (RSVD10):** Reserved

27.31 SSP Data (SSDR)—Offset 10h

The Enhanced SSP Data registers are single address locations that Read-Write data transfers can access. The SSDR represents two physical registers: the first is temporary storage for data on its way out through the Transmit FIFO, the other is temporary storage for data coming in through the Receive FIFO. As the system accesses the register, FIFO Control logic transfers data automatically between register and FIFO as fast as the system moves it. Data in the FIFO shifts up or down to accommodate the new word (unless it's an attempted Write to a full Transmit FIFO). Status bits (such as SSSR.TFL, SSSR.RFL, SSSR.RNE, and SSSR.TNF) show users whether the FIFO is full, above/below a programmable FIFO trigger threshold, or empty. For Transmit data transfers (Write from system to SSP peripheral), the register can be loaded (written) by the system processor anytime it falls below its threshold level when using programmed I/O. When a data size of less than 32-bits is selected, users should not left-justify data written to the Transmit FIFO. Transmit logic left-justifies the data and ignores any unused bits. Received data of less than 32-bits is automatically right-justified in the Receive FIFO. When the Enhanced SSP is programmed for National Semiconductor Microwire* frame format and if the size for Transmit data is 8-bits as selected by SSCR1.MWDS=0, then the most significant 24-bits are ignored. Similarly, if the size for the Transmit data is 16-bit as selected by SSCR1.MWDS=1, then most significant 16-bits are ignored. The SSCR0.DSS field controls the Receive data size.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**DATA (DATA):** Data word to be written to/read from transmit/ receive FIFO

27.32 SSP Time Out (SSTO)—Offset 28h

The Enhanced SSP Time-Out registers have single bit fields that specify the time-out value used to signal a period of inactivity within the Receive FIFO. Note that Writes to reserved bits must be zeroes, and Read value of these bits are undetermined.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h NA	**RSVD11 (RSVD11):** Reserved
23:0	0h RW	**TIMEOUT (TIMEOUT):** Timeout Value Is the value that defines the timeout interval, given by TIMEOUT/Peripheral Clock Frequency

27.33 reserved (SSPSP_res)—Offset 2Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**RSVD1 (RSVD1):** Reserved

27.34 reg_SSTSA_res (SSTSA_res)—Offset 30h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**RSVD0_31_0 (RSVD):** Reserved

27.35 SSRSA_Res (SSRSA_res)—Offset 34h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**RSVD1_32_0 (RSVD):** Reserved

27.36 SSTSS_res (SSTSS_res)—Offset 38h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RSVD (RSVD):** Reserved

27.37 (SSACD_res)—Offset 3Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**RSVD (RSVD):** Reserved

27.38 Reserved (ITF_res)—Offset 40h

Reserved

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h NA	**RSVD (RSVD):** Reserved

27.39 SPI Transmit FIFO (SITF)—Offset 44h

The SPI Transmit FIFO register is for writing the water mark for the SPI transmit FIFO and also for reading the number of entries in the SPI transmit FIFO

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h NA	**RSVD0_31_22 (RSVD0_31_22):** Reserved
21:16	0h RO	**SITFL (SITFL):** SPI Transmit FIFO Level Number of entries in SPI Transmit FIFO.
15:14	0h NA	**RSVD0_15_14 (RSVD0_15_14):** Reserved
13:8	0h RW	**LWMTF (LWMTF):** Low Water Mark Transmit FIFO. Set the low water mark of the SPI transmit FIFO.
7:6	0h NA	**RSVD0_7_6 (RSVD0_7_6):** Reserved
5:0	0h RW	**HWMTF (HWMTF):** High Water Mark Transmit FIFO. Set the high water mark of the SPI transmit FIFO.

27.40 SPI Receive FIFO (SIRF)—Offset 48h

The SPI Receive FIFO register is for writing the water mark for the SPI receive FIFO and also for reading the number of entries in the SPI receive FIFO

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h NA	**RSVD_31_14 (RSVD_31_14):** Reserved
13:8	0h RO	**SIRFL (SIRFL):** SPI Receive FIFO Level Number of entries in SPI Receive FIFO.
7:6	0h NA	**RSVD_7_6 (RSVD):** Reserved
5:0	0h RW	**WMRF (WMRF):** Water Mark Receive FIFO. Set the water mark of the SPI receive FIFO.

27.41 TPM Reg 0 (TPM_REG_0)—Offset FED40000h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**TPM Reg 0 (TPMR0):** This register is for TPM accesses.

27.42 TPM Reg 1 (TPM_REG_1)—Offset FED40004h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**TPM Reg 1 (TPMR1):** This register is for TPM accesses.

27.43 TPM Reg 2 (TPM_REG_2)—Offset FED40008h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**TPM Reg 2 (TPMR2):** This register is for TPM accesses.

27.44 TPM Reg 3 (TPM_REG_3)—Offset FED4000Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**TPM Reg 3 (TPMR3):** This register is for TPM accesses.

27.45 TPM Reg 4 (TPM_REG_4)—Offset FED40010h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**TPM Reg 4 (TPMR4):** This register is for TPM accesses.

27.46 CSXE Flash Primary Region (CSXE_MEFPREG)—Offset 0h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**CSME Flash Primary Region Limit (PRL):** This specifies address bits 26:12 for the Primary Region Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG2.Region Limit.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	0h RO/V	**CSME Flash Primary Region Base (PRB):** This specifies address bits 26:12 for the Primary Region Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG2.Region Base.

27.47 Hardware Sequencing Flash Status and Control (CSXE_HSFSTS_CTL)—Offset 4h

Several hardware sequenced operations are not supported in slave-attach flash mode.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 2000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved.
29:24	0h RW	**Flash Data Byte Count (FDBC):** This field specifies the number of bytes to shift in or out during the data portion of the SPI cycle. The contents of this register are 0s based with 0b representing 1 byte and 3Fh representing 64 bytes. The number of bytes transferred is the value of this field plus 1. This field is ignored for the Block Erase command.
23:22	0h RW	**Reserved RW Scratch Pad (Reserved_SCRATCHPAD):** Scratch Pad bits that are R/W to be used during ECO.
21	0h RW	**Write Enable Type (WET):** 0: Use 06h as the write enable instruction 1: Use 50h as the write enable instruction Note: No supported flash devices require the 50h opcode to enable a non-volatile status register write. This bit is no longer required.
20:17	0h RW	**Flash Cycle (FCYCLE):** This field defines the Flash SPI cycle type generated to the FLASH when the FGO bit is set as defined below: 0h: Read (1 up to 64 bytes by setting FDBC) 1h: Reserved 2h: Write (1 up to 64 bytes by setting FDBC) 3h: 4k Block Erase 4h: 64k Sector erase 5h: Read SFDP 6h: Read JEDEC ID 7h: write status 8h: read status 9h: RPMC Op1 (only legal in slave-attach flash mode) Ah: RPMC Op2 (only legal in slave-attach flash mode) Bh: Reserved Ch: Reserved Dh: Reserved Eh: Reserved Fh: Reserved Flash controller hardware automatically inserts a write enable opcode prior to Write and erase operations. Hardware automatically polls for device not-busy using read status after write and erase operations. If the device does not support 64k erase size (or if it does not support SFDP) then only 4k is allowed. Note: If reserved '1' is programmed to this field, flash controller will handle it as if it is 0 (Read) (Preserves legacy behavior.) Other reserved operations, including the RPMC, set the FCERR bit and do not issue a cycle on the SPI bus.

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW/1S/V	**Flash Cycle Go (FGO):** A write to this register with a '1' in this bit initiates a request to the Flash SPI Arbiter to start a cycle. This register is cleared by hardware when the cycle is granted by the SPI arbiter to run the cycle on the SPI bus. When the cycle is complete, the FDONE bit is set. Software is forbidden to write to any register in the HSFLCTL register between the FGO bit getting set and the FDONE bit being cleared. Any attempt to violate this rule will be ignored by hardware. Hardware allows other bits in this register to be programmed for the same transaction when writing this bit to 1. This saves an additional memory write. This bit remains set until the transaction is granted by the SPI Controller's internal arbiter.
15	0h RW/L	**Flash Configuration Lock-Down (FLOCKDN):** When set to 1, those Flash Program Registers that are locked down by this FLOCKDN bit cannot be written. Once set to 1, this bit can only be cleared by a hardware reset.
14	0h RO/V	**Flash Descriptor Valid (FDV):** This bit is set to a 1 if the Flash Controller read the correct Flash Descriptor Signature. If the Flash Descriptor Valid bit is not 1, software cannot use the Hardware Sequencing registers, but must use the software sequencing registers. Any attempt to use the Hardware Sequencing registers will result in the FCERR bit being set.
13	1h RO/V	**Flash Descriptor Override Pin-Strap Status (FDOPSS):** This register indicates whether the flash controller is overriding descriptor permissions due to the Pin-Strap. Note: the register value is the inversion of the level sampled on the external pinstrap. 1: No override 0: The Flash Descriptor Overide strap is set
12:9	0h RO	**Reserved (RSVD_1):** Reserved.
8	0h RW/1C/V	**SAF ctype error (H_SAF_CE):** Hardware sets this bit to 1 when a transaction is returned from the eSPI controller with ctype error.
7	0h RO/V	**SAF Mode Active (H_SAF_MODE_ACTIVE):** 0 : indicates flash is attached directly to the PCH via the SPI bus 1 : indicates flash is attached to the EC/BMC and is accessed via tunneled eSPI commands
6	0h RW/1C/V	**SAF link Error (H_SAF_LE):** Hardware sets this bit to 1 when a transaction is returned from the eSPI channel with link error.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RO/V	**SPI Cycle In Progress (H_SCIP):** Hardware sets this bit when software sets the Flash Cycle Go (FGO) bit in the Hardware Sequencing Flash Control register. This bit remains set until the cycle completes on the SPI interface or the transaction is prevented due to any protection policy violation (descriptor, address range, protected region, etc). Hardware automatically sets and clears this bit so that software can determine when read data is valid and/or when it is safe to begin programming the next command. Software must only program the next command when this bit is 0.
4	0h RW/1C/V	**SAF Data length Error (H_SAF_DLE):** Hardware sets this bit to 1 when a transaction is returned from the eSPI channel with an incorrect data length.
3	0h RW/1C/V	**SAF Error (H_SAF_ERROR):** Hardware sets this bit to 1 when a transaction is requested that is not supported by slave-attached flash, e.g. read status.
2	0h RW/1C/V	**Access Error Log (H_AEL):** Hardware sets this bit to a 1 when an attempt was made to access the BIOS region using the direct access method or an access to the BIOS Program Registers that violated the security restrictions. This bit is simply a log of an access security violation. This bit is cleared by software writing a '1'.
1	0h RW/1C/V	**Flash Cycle Error (FCERR):** Hardware sets this bit to 1 when a program register access is blocked to the FLASH due to one of the protection policies or when any of the programmed cycle registers is written while a programmed access is already in progress. This bit remains asserted until cleared by software writing a 1 or until a partition reset occurs. Software must clear this bit before setting the FLASH Cycle GO bit in this register.
0	0h RW/1C/V	**Flash Cycle Done (FDONE):** The PCH sets this bit to 1 when the SPI Cycle completes after software previously set the FGO bit. This bit remains asserted until cleared by software writing a 1 or partition reset.

27.48 Flash Address (CSXE_FADDR)—Offset 8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:27	0h RO	**Reserved (RSVD):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
26:0	0h RW	**Flash Linear Address (FLA):** The FLA is the starting byte linear address of a SPI Read or Write cycle or an address within a Block for the Block Erase command. The Flash Linear Address must fall within a region for which CSXE has access permissions. Hardware must convert the FLA into a Flash Physical Address (FPA) before running this cycle on the SPI bus.

27.49 Discrete Lock Bits (CSXE_DLOCK)—Offset Ch

Lockable CSxE registers may be locked by either the global FLOCKDN bit or by the individual DLOCK.* bit. Each lockable bit in this register is locked either by itself or by the FLOCKDN bit.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RSVD):** Reserved.
16	0h RW/L	**SSEQ Lock-Down (SSEQLOCKDN):** CSxE Software Sequencing registers are locked when the logical OR of this bit and FLOCKDN is true. The affected registers are SSFSTS_CTL.SCF, PREOP_OPTYPE, OPMENU0, and OPMENU1.**Once set to 1 this register is only cleared by CSxE partition reset.**

27.50 Flash Data 0 (CSXE_FDATA0)—Offset 10h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 0 (FD0):** This field is shifted out as the SPI Data on the Master-Out Slave-In Data pin during the data portion of the SPI cycle. This register also shifts in the data from the Master-In Slave-Out pin into this register during the data portion of the SPI cycle. The data is always shifted starting with the least significant byte, msb to lsb, followed by the next least significant byte, msb to lsb, etc. Specifically, the shift order on SPI in terms of bits within this register is: 7-6-5-4-3-2-1-0-15-14-13- ... 8-23-22- ... 16-31 ... 24 Bit 24 is the last bit shifted out/in. There are no alignment assumptions; byte 0 always represents the value specified by the cycle address. Note that the data in this register may be modified by the hardware during any programmed SPI transaction. Direct Memory Reads do not modify the contents of this register.

27.51 Flash Data 1 (CSXE_FDATA1)—Offset 14h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 1 (FD1):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD0 has completely shifted in/out.

27.52 Flash Data 2 (CSXE_FDATA2)—Offset 18h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 2 (FD2):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD1 has completely shifted in/out.

27.53 Flash Data 3 (CSXE_FDATA3)—Offset 1Ch

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 3 (FD3):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD2 has completely shifted in/out.

27.54 Flash Data 4 (CSXE_FDATA4)—Offset 20h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 4 (FD4):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD3 has completely shifted in/out.

27.55 Flash Data 5 (CSXE_FDATA5)—Offset 24h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 5 (FD5):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD4 has completely shifted in/out.

27.56 Flash Data 6 (CSXE_FDATA6)—Offset 28h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 6 (FD6):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD5 has completely shifted in/out.

27.57 Flash Data 7 (CSXE_FDATA7)—Offset 2Ch

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 7 (FD7):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD6 has completely shifted in/out.

27.58 Flash Data 8 (CSXE_FDATA8)—Offset 30h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 8 (FD8):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD7 has completely shifted in/out.

27.59 Flash Data 9 (CSXE_FDATA9)—Offset 34h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 9 (FD9):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD8 has completely shifted in/out.

27.60 Flash Data 10 (CSXE_FDATA10)—Offset 38h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 10 (FD10):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD9 has completely shifted in/out.

27.61 Flash Data 11 (CSXE_FDATA11)—Offset 3Ch

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 11 (FD11):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD10 has completely shifted in/out.

27.62 Flash Data 12 (CSXE_FDATA12)—Offset 40h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 12 (FD12):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD11 has completely shifted in/out.

27.63 Flash Data 13 (CSXE_FDATA13)—Offset 44h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 13 (FD13):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD12 has completely shifted in/out.

27.64 Flash Data 14 (CSXE_FDATA14)—Offset 48h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 14 (FD14):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD13 has completely shifted in/out.

27.65 Flash Data 15 (CSXE_FDATA15)—Offset 4Ch

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 15 (FD15):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD14 has completely shifted in/out.

27.66 Flash Region Access Permissions (CSXE_FRACC)—Offset 50h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:404h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW/L	**CSME Master Write Access Grant (MEMWAG):** Each bit [31:24] corresponds to Master[7:0]. CSME can grant one or more masters write access to the CSxE region 2 overriding the permissions in the Flash Descriptor.
23:16	0h RW/L	**CSME Master Read Access Grant (MEMRAG):** Each bit [23:16] corresponds to Master[7:0]. CSME can grant one or more masters read access to the CSME region 2 overriding the read permissions in the Flash Descriptor.
15:8	4h RO/V	**CSME Region Write Access (MERWA):** Each bit [15:8] corresponds to Regions [7:0]. If the bit is set, this master can erase and write that particular region through register accesses. The contents of this register are that of the Flash Descriptor.Flash Master 2.Master Region Write Access OR a particular master has granted CSME write permissions in their Master Write Access Grant register OR the Flash Descriptor Security Override strap is set. CSME always have the write access to its own Region 2 by default. See also CM_WAP
7:0	4h RO/V	**CSME Region Read Access (MERRA):** Each bit [7:0] corresponds to Regions [7:0]. If the bit is set, this master can read that particular region through register accesses. The contents of this register are that of the Flash Descriptor.Flash Master 2.Master Region Read Access OR a particular master has granted CSME read permissions in their Master Read Access Grant register OR the Flash Descriptor Security Override strap is set. CSME always have the read access to its own Region 2 by default. See also CM_RAP

27.67 Flash Region 0 (CSXE_FREG0)—Offset 54h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 0 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG0.Region Limit.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	0h RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 0 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG0.Region Base.

27.68 Flash Region 1 (CSXE_FREG1)—Offset 58h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 1 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG1.Region Limit.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 1 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG1.Region Base.

27.69 Flash Region 2 (CSXE_FREG2)—Offset 5Ch

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 2 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG2.Region Limit.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 2 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG2.Region Base.

27.70 Flash Region 3 (CSXE_FREG3)—Offset 60h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 3 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG3.Region Limit.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 3 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG3.Region Base.

27.71 Flash Region 4 (CSXE_FREG4)—Offset 64h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 4 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG4.Region Limit.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 4 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG4.Region Base.

27.72 Flash Region 5 (CSXE_FREG5)—Offset 68h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 5 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG5.Region Limit.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 5 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG5.Region Base.

27.73 Flash Region 6 (CSXE_FREG6)—Offset 6Ch

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 6 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG6.Region Limit if the Server Feature Disable Fuse 6 is '0'. If the Server Feature Disable Fuse 6 is a '1' then this field is loaded with all zeros.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 6 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG6.Region Base if the Server Feature Disable Fuse 6 is '0'. If the Server Feature Disable Fuse 6 is a '1' then this field is loaded with all ones.

27.74 Flash Region 7 (CSXE_FREG7)—Offset 70h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 7 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG7.Region Limit.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 7 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG7.Region Base.

27.75 Flash Region 8 (CSXE_FREG8)—Offset 74h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 8 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG8.Region Limit.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 8 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG8.Region Base.

27.76 Flash Region 9 (CSXE_FREG9)—Offset 78h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 9 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG9.Region Limit.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 9 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG9.Region Base.

27.77 Flash Region 10 (CSXE_FREG10)—Offset 7Ch

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 10 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG10.Region Limit
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 10 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG10.Region Base

27.78 Flash Region 11 (CSXE_FREG11)—Offset 80h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 11 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG11.Region Limit
15	0h RO	**Reserved (RSVD_1):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 11 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG11.Region Base

27.79 Flash Protected Range 0 (CSXE_FPR0)—Offset 84h

This register cannot be written when the FLOCKDN bit is set to 1.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/L	**Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
30:16	0h RW/L	**Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range. Address bits 11:0 are assumed to be FFFh for the limit comparison. Any address greater than the value programmed in this field is unaffected by this protected range.
15	0h RW/L	**Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that reads directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
14:0	0h RW/L	**Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range. Address bits 11:0 are assumed to be 000h for the base comparison. Any address less than the value programmed in this field is unaffected by this protected range.

27.80 Flash Protected Range 1 (CSXE_FPR1)—Offset 88h

This register can not be written when the FLOCKDN bit is set to 1.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/L	**Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
30:16	0h RW/L	**Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range. Address bits 11:0 are assumed to be FFFh for the limit comparison. Any address greater than the value programmed in this field is unaffected by this protected range.
15	0h RW/L	**Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that reads directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
14:0	0h RW/L	**Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range. Address bits 11:0 are assumed to be 000h for the base comparison. Any address less than the value programmed in this field is unaffected by this protected range.

27.81 Flash Protected Range 2 (CSXE_FPR2)—Offset 8Ch

This register can not be written when the FLOCKDN bit is set to 1.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/L	**Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
30:16	0h RW/L	**Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range. Address bits 11:0 are assumed to be FFFh for the limit comparison. Any address greater than the value programmed in this field is unaffected by this protected range.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/L	**Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that reads directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
14:0	0h RW/L	**Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range. Address bits 11:0 are assumed to be 000h for the base comparison. Any address less than the value programmed in this field is unaffected by this protected range.

27.82 Flash Protected Range 3 (CSXE_FPR3)—Offset 90h

This register cannot be written when the FLOCKDN bit is set to 1.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/L	**Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
30:16	0h RW/L	**Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range. Address bits 11:0 are assumed to be FFFh for the limit comparison. Any address greater than the value programmed in this field is unaffected by this protected range.
15	0h RW/L	**Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that reads directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
14:0	0h RW/L	**Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range. Address bits 11:0 are assumed to be 000h for the base comparison. Any address less than the value programmed in this field is unaffected by this protected range.

27.83 Flash Protected Range 4 (CSXE_FPR4)—Offset 94h

This register can not be written when the FLOCKDN bit is set to 1.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/L	**Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
30:16	0h RW/L	**Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range. Address bits 11:0 are assumed to be FFFh for the limit comparison. Any address greater than the value programmed in this field is unaffected by this protected range.
15	0h RW/L	**Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that reads directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
14:0	0h RW/L	**Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range. Address bits 11:0 are assumed to be 000h for the base comparison. Any address less than the value programmed in this field is unaffected by this protected range.

27.84 Write Protected Range 0 (CSXE_WPR0)—Offset 98h

The region defined by this register is neither writable nor erasable by any master. The writes/erases are blocked regardless of access method, e.g. hardware sequencing, software sequencing, IOSF sideband message, etc. In contrast to the CSME's Protected Range registers which only prevent the CSME from accessing the Protected Range, the Blocked Range register blocks access by all masters. If the hardware detects an illegal access to the Write Protected Range, it shall not perform the write/erase to the flash device. Error reporting and completion of the transaction is the same as for an illegal Protected Range access. In response to an IOSF sideband SpiWrite or SpiErase to the Write Protected Range, the Flash Controller shall return a Completion message with Unsupported Request without sending the transaction to the flash device. This register cannot be written when the CSME Flash Program Register HSFSTS.FLOCKDN bit is set to 1.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/L	**Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
30:16	0h RW/L	**Blocked Range Limit (BRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range. Address bits 11:0 are assumed to be FFFh for the limit comparison. Any address greater than the value programmed in this field is unaffected by this protected range.
15	0h RO	**Reserved (RSVD):** Reserved.
14:0	0h RW/L	**Blocked Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range. Address bits 11:0 are assumed to be 000h for the base comparison. Any address less than the value programmed in this field is unaffected by this protected range.

27.85 Software Sequencing Flash Status and Control (CSXE_SSFSTS_CTL)—Offset A0h

The Software Sequencing control and status registers are intended to be used only as a back-up mode to the hardware sequencing control and status registers.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:FE000000h

Bit Range	Default & Access	Field Name (ID): Description
31:27	1Fh RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
26:24	6h RW/L	**SPI Cycle Frequency (SCF):** The listed frequencies are approximate. 000: 120MHz (not supported in SPT) 001: 60MHz (not supported in SPT) 010: 48MHz 011: 40MHz (not supported in SPT) 100: 30MHz 101: 24MHz (not supported in SPT) 110: 17MHz All others: Reserved This register sets frequency to use for all SPI Software Sequencing cycles (write, erase, fast read, read status, etc) except for the Read cycle which always run at 20MHz. This register is locked when the SPI Configuration Lock-Down (FLOCKDN) bit is set. Note: Fast Reset Testmode overrides the value programmed into this register.
23	0h RO	**Reserved (SME):** Reserved. Was previously the SMI Enable field
22	0h RW	**Data Cycle (DS):** When set to 1, there is data that corresponds to this transaction. When 0, no data is delivered for this cycle, and the DBC and data fields themselves are do not care.
21:16	0h RW	**Data Byte Count (DBC):** This field specifies the number of bytes to shift in or out during the data portion of the SPI cycle. The valid settings (in decimal) are any value from 0 to 63. The number of bytes transferred is the value of this field plus 1. Note that when this field is 00_0000b, then there is 1 byte to transfer and that 11_1111b means there are 64 bytes to transfer.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:12	0h RW	**Cycle Opcode Pointer (COP):** This field selects one of the programmed opcodes in the Opcode Menu to be used as the SPI Command/Opcode. In the case of an Atomic Cycle Sequence, this determines the second command.
11	0h RW	**Sequence Prefix Opcode Pointer (SPOP):** This field selects one of the two programmed prefix opcodes for use when performing an Atomic Cycle Sequence. A value of 0 points to the opcode in the least significant byte of the Prefix Opcodes register. By making this programmable, the ICH supports flash devices that have different opcodes for enabling writes to the data space vs. status register.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW	**Atomic Cycle Sequence (ACS):** When set to 1 along with the SCGO assertion, the PCH will execute a sequence of commands on the SPI interface without allowing another master to arbitrate and interleave cycles. The sequence is composed of: - Atomic Sequence Prefix Command (8-bit opcode only) - Primary Command specified below by software (can include address and data) - Polling the Flash Status Register (opcode 05h) until bit 0 becomes 0b. The SPI Cycle in Progress bit remains set and the Cycle Done Status bit remains unset until the Busy bit in the Flash Status Register returns 0.
9	0h RW/1S/V	**SPI Cycle Go (SCGO):** A write to this register with a '1' in this bit starts the SPI cycle defined by the other bits of this register. The SPI Cycle in Progress (SCIP) bit gets set by this action. Hardware must ignore writes to this bit while the Cycle In Progress bit is set. Hardware allows other bits in this register to be programmed for the same transaction when writing this bit to 1. This saves an additional memory write. This bit remains set until the transaction is granted by the SPI Controller's internal arbiter.
8	0h RO	**Reserved (RSVD_2):** Reserved.
7	0h RO/V	**Fast Read Supported (FAST_READ_SUPPORT):** This bit reflects the value of the Fast Read Support bit in the Flash Descriptor Component Section.
6	0h RO/V	**Dual Output Fast Read Supported (DUAL_FAST_READ_SUPPORT):** This bit reflects the value of the Dual Output Fast Read Support bit in the Flash Descriptor Component Section.
5	0h RO	**Reserved (RSVD_3):** Reserved.
4	0h RO/V	**Access Error Log (AEL):** This bit reflects the value of the Hardware Sequencing Status.AEL register.
3	0h RW/1C/V	**Flash Cycle Error (FCERR):** Hardware sets this bit to 1 when a programmed access is blocked from running on the SPI interface due to one of the protection policies or when any of the programmed cycle registers is written while a programmed access is already in progress. This bit remains asserted until cleared by software writing a 1 or partition reset.
2	0h RW/1C/V	**Cycle Done Status (CYCLE_DONE_STS):** The PCH sets this bit to 1 when the SPI Cycle completes (i.e., SCIP bit is 0) after software sets the GO bit. This bit remains asserted until cleared by software writing a 1 or hardware reset. When this bit is set and the SPI SMI# Enable bit is set, an internal signal is asserted to the SMI# generation block. Software must make sure this bit is cleared prior to enabling the SPI SMI# assertion for a new programmed access.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RO	**Reserved (RSVD_4):** Reserved.
0	0h RO/V	**SPI Cycle In Progress (SCIP):** Hardware sets this bit when software sets the SPI Cycle Go bit in the Command register. This bit remains set until the cycle completes on the SPI interface or the transaction is prevented due to any protection policy violation (descriptor, address range, protected region, etc). Hardware automatically sets and clears this bit so that software can determine when read data is valid and/or when it is safe to begin programming the next command. Software must only program the next command when this bit is 0.

27.86 Prefix Opcode and Opcode Type Configuration (CSXE_PREOP_OPTYPE)—Offset A4h

This register is not writable when the Flash Configuration Lock-Down (FLOCKDN) bit is set. Entries in this register correspond to the entries in the Opcode Menu Configuration register. Note that the definition below only provides write protection for opcodes that have addresses associated with them. Therefore, any erase or write opcodes that do not use an address should be avoided (for example, 'Chip Erase' and 'Auto-Address Increment Byte Program').

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RW/L	**Opcode Type 7 (OPCODE_TYPE7):** See the description for bits 1:0.
29:28	0h RW/L	**Opcode Type 6 (OPCODE_TYPE6):** See the description for bits 1:0.
27:26	0h RW/L	**Opcode Type 5 (OPCODE_TYPE5):** See the description for bits 1:0.
25:24	0h RW/L	**Opcode Type 4 (OPCODE_TYPE4):** See the description for bits 1:0.
23:22	0h RW/L	**Opcode Type 3 (OPCODE_TYPE3):** See the description for bits 1:0.
21:20	0h RW/L	**Opcode Type 2 (OPCODE_TYPE2):** See the description for bits 1:0.
19:18	0h RW/L	**Opcode Type 1 (OPCODE_TYPE1):** See the description for bits 1:0.

Bit Range	Default & Access	Field Name (ID): Description
17:16	0h RW/L	**Opcode Type 0 (OPCODE_TYPE0):** This field specifies information about the corresponding Opcode 0. This information allows the hardware to 1) know whether to use the address field and 2) provide BIOS and Shared Flash protection capabilities. The hardware implementation also uses the read vs. write information for modifying the behavior of the SPI interface logic. The encoding of the two bits is: 00: No Address associated with this Opcode and Read Cycle type 01: No Address associated with this Opcode and Write Cycle type 10: Address required; Read cycle type 11: Address required; Write cycle type
15:8	0h RW/L	**Prefix Opcode 1 (PREFIX_OPCODE1):** Software programs an SPI opcode into this field that is permitted to run as the first command in an atomic cycle sequence.
7:0	0h RW/L	**Prefix Opcode 0 (PREFIX_OPCODE0):** Software programs an SPI opcode into this field that is permitted to run as the first command in an atomic cycle sequence.

27.87 Opcode Menu0 Configuration (CSXE_OPMENU0)—Offset A8h

This register is not writable when the SPI Configuration Lock-Down bit (FLOCKDN) is set. Eight entries are available in this register to give BIOS a sufficient set of commands for communicating with the flash device, while also restricting what malicious software can do. This keeps the hardware flexible enough to operate with a wide variety of SPI devices. It is recommended that BIOS avoid programming Write Enable opcodes in this menu. Malicious software could then perform writes and erases to the SPI flash without using the atomic cycle mechanism. This could cause functional failures in a shared flash environment. Write Enable opcodes should only be programmed in the Prefix Opcodes.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW/L	**Allowable Opcode 3 (OPCODE3):** See the description for bits 7:0.
23:16	0h RW/L	**Allowable Opcode 2 (OPCODE2):** See the description for bits 7:0.
15:8	0h RW/L	**Allowable Opcode 1 (OPCODE1):** See the description for bits 7:0.
7:0	0h RW/L	**Allowable Opcode 0 (OPCODE0):** Software programs an SPI opcode into this field for use when initiating SPI commands through the Control Register.

27.88 Opcode Menu1 Configuration (CSXE_OPMENU1)—Offset ACh

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW/L	**Allowable Opcode 7 (OPCODE7):** See the description in OPMENU0.
23:16	0h RW/L	**Allowable Opcode 6 (OPCODE6):** See the description in OPMENU0.
15:8	0h RW/L	**Allowable Opcode 5 (OPCODE5):** See the description in OPMENU0.
7:0	0h RW/L	**Allowable Opcode 4 (OPCODE4):** See the description in OPMENU0.

27.89 Vendor Specific Component Capabilities for Component 0 (CSXE_SFDP0_VSCC0)—Offset C4h

The fields in this register pertain to cycles targeting addresses within Component 0. The lockable bits in this register are locked when either CPPTV is set to '1' by hardware or when VCL is '1'.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:2000h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Component Property Parameter Table Valid (CPPTV):** This bit is set to a 1 if the Flash Controller detects a valid SFDP Component Property Parameter Table in Component 0. Note: If this bit is set software must not overwrite bits that were initialized by hardware via SFDP discovery.
30	0h RW/L	**Vendor Component Lock (VCL)**
29	0h RW/V/L	**64k Erase Valid (EO_64k_VALID)**
28	0h RW/V/L	**4k Erase Valid (EO_4k_VALID)**
27	0h RW/L	**RPMC Supported (RPMC_SUPPORTED)**

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW/V/L	**Deep Powerdown Supported (DEEP_PWRDN_SUPPORTED)**
25	0h RW/V/L	**Suspend/Resume Supported (SUSPEND_RESUME_SUPPORTED)**
24	0h RW/V/L	**Soft Reset Supported (SOFT_RST_SUPPORTED)**
23:16	0h RW/V/L	**64k Erase Opcode (EO_64k)**
15:8	20h RW/V/L	**4k Erase Opcode (EO_4k)**
7:5	0h RW/V/L	**Quad Enable Requirements (QER)**
4	0h RW/V/L	**Write Enable on Write Status (WEWS)**
3	0h RW/V/L	**Write Status Required (WSR)**
2	0h RW/V/L	**Write Granularity (WG)**
1:0	0h RW/L	**Reserved (RSVD)**

27.90 Vendor Specific Component Capabilities for Component 1 (CSXE_SFDP1_VSCC1)—Offset C8h

The fields in this register pertain to cycles targeting addresses outside of Component 0. The lockable bits in this register are locked when either CPPTV is set to '1' by hardware or when VCL is '1'.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:2000h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Component Property Parameter Table Valid (CPPTV):** This bit is set to a 1 if the Flash Controller detects a valid SFDP Component Property Parameter Table in Component 1. Note: If this bit is set software must not overwrite bits that were initialized by hardware via SFDP discovery.
30	0h RO	**Reserved (RSVD)**
29	0h RW/V/L	**64k Erase Valid (EO_64k_VALID)**
28	0h RW/V/L	**4k Erase Valid (EO_4k_VALID)**

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW/L	**RPMC Supported (RPMC_SUPPORTED):** 0 the device does not support RPMC 1 the device supports RPMC
26	0h RW/V/L	**Deep Powerdown Supported (DEEP_PWRDN_SUPPORTED)**
25	0h RW/V/L	**Suspend/Resume Supported (SUSPEND_RESUME_SUPPORTED)**
24	0h RW/V/L	**Soft Reset Supported (SOFT_RST_SUPPORTED)**
23:16	0h RW/V/L	**64k Erase Opcode (EO_64k)**
15:8	20h RW/V/L	**4k Erase Opcode (EO_4k)**
7:5	0h RW/V/L	**Quad Enable Requirements (QER)**
4	0h RW/V/L	**Write Enable on Write Status (WEWS)**
3	0h RW/V/L	**Write Status Required (WSR)**
2	0h RW/V/L	**Write Granularity (WG)**
1:0	0h RW/L	**Reserved (RSVD_1)**

27.91 Parameter Table Index (CSXE_PTINX)—Offset CCh

Observability control for Component Property Tables. Note: The PTINX and PTDATA registers do not have any meaning in slave-attach flash mode because the SPI controller does not perform SFDP discovery.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved.
15:14	0h RW	**Supported Parameter Table (SPT):** Selects which supported parameter table to observe. 00: Component 0 Property Parameter Table. 01: Component 1 Property Parameter Table. 10 - 11: Reserved.

Bit Range	Default & Access	Field Name (ID): Description
13:12	0h RW	**Header or Data (HORD):** Select parameter table header DW vs Data DW. 00: SFDP Header. 01: Parameter Table Header. 10: Data. 11: Reserved.
11:2	0h RW	**Parameter Table DW Index (PTDWI):** Selects the DW offset within the parameter table to observe. The returned data is undefined if the index is programmed to a value greater than the size of the header or table.
1:0	0h RO	**Reserved (RSVD_1):** Reserved.

27.92 Parameter Table Data (CSXE_PTDATA)—Offset D0h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Parameter Table DW Data (PTDWD):** Returns the DW of data to observe as selected in the Parameter Table Index register. Note: The returned value of reserved fields in the SFDP header or table must be ignored by software. The flash controller may return either 0 or 1 for these fields.

27.93 SPI Bus Requester Status (CSXE_SBRS)—Offset D4h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**TPM Access Ongoing (TPM_ACC_ONG)**
30	0h RO/V	**eSPI Access Ongoing (ESPI_ACC_ONG):** This bit is only defined if eSPI and SPI are sharing the SPI bus.
29	0h RO/V	**Touch Access Ongoing (TOUCH_ACC_ONG)**

Bit Range	Default & Access	Field Name (ID): Description
28	0h RO/V	**CSME accessing the Huffman Decompression (CSME_ACC_HWD)**
27:18	0h RO	**Reserved (RSVD):** Reserved
17:15	0h RO/V	**Master 5 Status (M5STATUS):** See description under M1STATUS Note: Master numbering was changed in a late revision of the Cspec, after this register was already implemented in RTL. Hence the master numbering in this register is not consecutive.
14:12	0h RO/V	**Master 6 Status (M6STATUS):** See description under M1STATUS.
11:9	0h RO/V	**Master 4 Status (M4STATUS):** See description under M1STATUS.
8:6	0h RO/V	**Master 3 Status (M3STATUS):** See description under M1STATUS.
5:3	0h RO/V	**Master 2 Status (M2STATUS):** See description under M1STATUS.
2:0	0h RO/V	**Master 1 Status (M1STATUS):** Indicates whether this master has an outstanding transaction enqueued or in flight and the transaction type. 0xx: No transaction. 100: Flash read transaction. 101: Flash write transaction. 110: Flash erase transaction. 111: Flash RPMC transaction.

27.94 Huffman Decompression Compressed Page Offset (CSXE_HDCOMPOFF)—Offset D8h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:27	0h RO	**Reserved (RSVD):** Reserved.
26:0	0h RW	**Compressed Offset (COMPOFF):** Hardware adds this field to the address read from the LUT to calculate a page's start address.

27.95 Touch Controller Configuration Register (CSXE_TCCR)—Offset DCh

Touch devices are required to support single IO fast read 0Bh, therefore no control bits are allocated in this register. Timing is always 8 dummy clocks, mode bits are not supported, and the opcodes are fixed to de-facto standards from the flash industry. Firmware is required to enforce the Touch Maximum Frequency softstrap when programming this register; no hardware checks are done.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:60h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**Reserved (RSVD):** Reserved.
6:4	6h RW	**Touch Cycle Frequency (TCF):** The listed frequencies are approximate. 000 : 120MHz (not supported) 001 : 60MHz (not supported) 010 : 48MHz, divide by 2.5 011 : 40MHz (not supported) 100 : 30MHz, divide by 4 101 : 24MHz, divide by 5 (supported by some SOCs) 110 : 17MHz, divide by 7 111 : Reserved
3	0h RW	**Quad IO Read (EBh) enabled (QIORE):** Firmware programs this bit after discovering capabilities of Touch device. 0 : not enabled 1 : enabled
2	0h RW	**Dual IO Read (BBh) enabled (DIORE):** Firmware programs this bit after discovering capabilities of Touch device. 0 : not enabled 1 : enabled
1	0h RW	**Reserved (RSVD_1):** Reserved.
0	0h RO/V	**Touch Cycle Busy (TCB):** This bit is '1' if and only if the flash controller is busy with a touch transaction. When this bit is '0' the SPI controller is ready to accept another write transaction from firmware.

27.96 Flash Region 12 (CSXE_FREG12)—Offset E0h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 12 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG12.Region Limit
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 12 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG12.Region Base

27.97 Flash Region 13 (CSXE_FREG13)—Offset E4h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 13 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG13.Region Limit
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 13 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG13.Region Base

27.98 Flash Region 14 (CSXE_FREG14)—Offset E8h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 14 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG14.Region Limit
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 14 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG14.Region Base

27.99 Flash Region 15 (CSXE_FREG15)—Offset ECh

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 15 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG15.Region Limit
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 15 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG15.Region Base

27.100 Interrupt Cause Enable (CSXE_ICE)—Offset F0h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW/V/P	**IE Software Sequencing Enable (IESSEN):** <product specific = [quote]DNV, LBG[/quote]> This bit is only implemented in DNV, LBG and later SOCs. This bit controls whether the IE software sequencing 'go' bit can be set by the IE. The initial value of this bit is loaded from softstrap spi_ie_ss_enable_default. 0 : 'go' bit in IE software sequencing register is read-only 1 : 'go' bit in IE software sequencing register is writable
10:9	0h RO	**Reserved (RSVD_4):** Reserved.
8	0h RW/V/P	**Host Software Sequencing Enable (HSSEN):** This bit controls whether the host software sequencing 'go' bit can be set by the host. The initial value of this bit is loaded from softstrap spi_host_ss_enable_default. 0 : 'go' bit in host software sequencing register is read-only 1 : 'go' in host software sequencing register is writable
7	0h RO	**Reserved (RSVD_3):** Reserved.
6	0h RO/V	**Server Features Disable Fuse 6 Shadow (SFDF6):** This bit reflects the state of the Server Features Disable fuse 6 input to the SPI controller.
5:3	0h RO	**Reserved (RSVD_1):** Reserved.
2	0h RO	**Reserved (RSVD_2):** Reserved.
1	0h RW	**Software Sequencing Interrupt Enable (SSIE):** This functionality is not supported by the SPI flash controller. Setting this bit to 1 may cause undefined behavior. Note: CSxE firmware does not use this feature.
0	0h RW	**Hardware Sequencing Interrupt Enable (HSIE):** This functionality is not supported by the SPI flash controller. Setting this bit to 1 may cause undefined behavior. Note: CSxE firmware does not use this feature.

27.101 Interrupt Cause Status (CSXE_ICS)—Offset F4h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RO	**Reserved (RSVD):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW/1C/V	**Software Sequencing Interrupt Status (SSIS):** This register is set to '1' whenever the CSxE Software Sequencing Cycle Done status register is set to a '1'. This register is cleared when firmware writes a '1' to it.
0	0h RW/1C/V	**Hardwareware Sequencing Interrupt Status (HSIS):** This register is set to '1' whenever the CSxE Hardware Sequencing Flash Cycle Done status register is set to a '1'. This register is cleared when firmware writes a '1' to it.

27.102 Device 0 Write/Erase Count (CSXE_D0WEC)—Offset FCh

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD)**
30:0	0h RW/V	**Device 0 Write/Erase Count (D0WEC):** Incremented by the flash controller on every write or erase to device 0, regardless of which master initiated the transaction.

27.103 Device 1 Write/Erase Count (CSXE_D1WEC)—Offset 100h

This functionality is not implemented in the SIP

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD)**
30:0	0h RW/V	**Device 1 Write/Erase Count (D1WEC):** Incremented by the flash controller on every write or erase to device 1, regardless of which master initiated the transaction.

27.104 Write/Erase Count Threshold (CSXE_WECTHR)—Offset 104h

This functionality is not implemented in the SIP

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**WEC Interrupt Enable (WEC_IE):** 0 : Write/Erase count interrupts to the CSME are disabled 1 : Write/Erase count interrupts to the CSME are enabled **Note:** This functionality is not supported by the SPI flash controller. Setting this bit to 1 may cause undefined behavior.
30:0	0h RW	**Write/Erase Count Threshold (WEC_THR):** Threshold for generating a wear-out attack interrupt to the CSME.

27.105 RPMC SFDP Table (CSXE_RPMC0_D0)—Offset 108h

The SPI controller hardware does not read the RPMC tables from SFDP. If RPMC operations are desired via Software Sequencing, CSxE firmware must at a minimum program the RPMC opcodes supported by the flash device. All other bits of this register are not used by the SPI controller hardware in any version of the SIP. Mapping the registers into CSxE MMIO space allows the option of firmware discovering RPMC rather than hardware.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:969B00h

Bit Range	Default & Access	Field Name (ID): Description
31:0	969B00h RW	**RPMC SFDP Table dword 0 Device 0 (RPMC0_D0):** See definition in the RPMC specification. http://downloadcenter.intel.com/Detail_Desc.aspx?agr=Y&DwnldID=22646 Bits 15:8 = RPMC OP1 Opcode Bits 23:16 = RPMC OP2 Opcode During software sequenced cycles from CSxE, the SPI controller hardware will compare the SW sequenced opcode with the OP1 and OP2 opcodes programmed here to determine if an RPMC cycle is being attempted via software sequencing.

27.106 RPMC SFDP Table (CSXE_RPMC1_D0)—Offset 10Ch

This RPMC register is not used by the SPI controller hardware in any version of the SIP. Mapping this register into CSxE MMIO space allows the option of firmware discovering RPMC rather than hardware.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**RPMC SFDP Table dword 1 Device 0 (RPMC1_D0):** See definition in the RPMC specification. http://downloadcenter.intel.com/Detail_Desc.aspx?agr=Y&DwnldID=22646

27.107 RPMC SFDP Table (CSXE_RPMC0_D1)—Offset 110h

The SPI controller hardware does not read the RPMC tables from SFDP. If RPMC operations are desired via Software Sequencing, CSxE firmware must at a minimum program the RPMC opcodes supported by the flash device. All other bits of this register are not used by the SPI controller hardware in any version of the SIP. Mapping the registers into CSxE MMIO space allows the option of firmware discovering RPMC rather than hardware.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:969B00h

Bit Range	Default & Access	Field Name (ID): Description
31:0	969B00h RW	**RPMC SFDP Table dword 0 Device 1 (RPMC0_D1):** See definition in the RPMC specification. http://downloadcenter.intel.com/Detail_Desc.aspx?agr=Y&DwnldID=22646 Bits 15:8 = RPMC OP1 Opcode Bits 23:16 = RPMC OP2 Opcode During software sequenced cycles from CSxE, the SPI controller hardware will compare the SW sequenced opcode with the OP1 and OP2 opcodes programmed here to determine if an RPMC cycle is being attempted via software sequencing.

27.108 RPMC SFDP Table (CSXE_RPMC1_D1)—Offset 114h

This RPMC register is not used by the SPI controller hardware in any version of the SIP. Mapping this register into CSxE MMIO space allows the option of firmware discovering RPMC rather than hardware.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**RPMC SFDP Table dword 1 Device 1 (RPMC1_D1):** See definition in the RPMC specification. http://downloadcenter.intel.com/Detail_Desc.aspx?agr=Y&DwnldID=22646

27.109 CSME Master Read Access Permissions (CSXE_CM_RAP)—Offset 118h

CM_RAP, CM_RAP registers are implemented in DNV, LBG and subsequent programs.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:4h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved.
15:0	4h RO/V	**CSME Master Read Access Pemissions (CMRAP) (CMRAP):** Each bit [15:0] corresponds to Regions [15:0]. If the bit is set, this master can read that particular region through register accesses. The contents of this register are that of the Flash Descriptor.Flash Master 2.Master Region Read Access OR a particular master has granted CSME read permissions in their Master Read Access Grant register OR the Flash Descriptor Security Override strap is set. CSME has read access to its own Region 2 by default. Thus, the reset default of this field will be read as [quote]4h[/quote].

27.110 CSME Master Write Access Permissions (CSXE_CM_WAP)—Offset 11Ch

CM_RAP, CM_RAP registers are implemented in DNV, LBG and subsequent programs.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:4h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved.
15:0	4h RO/V	**CSME Master Write Access Pemissions (CMRAP) (CMWAP):** Each bit [15:0] corresponds to Regions [15:0]. If the bit is set, this master can erase and write that particular region through register accesses. The contents of this register are that of the Flash Descriptor.Flash Master 2.Master Region Write Access OR a particular master has granted CSME write permissions in their Master Write Access Grant register OR the Flash Descriptor Security Override strap is set. CSME has write access to its own Region 2 by default. Thus, the reset default of this field will be read as [quote]4h[/quote].

27.111 Host Flash Protected Range 0 (CSXE_HOST_PR0)—Offset 184h

These are not physical registers in the CSME memory space. These addresses provide the CSME with a read-only view of the values stored in the Host Protected Range registers.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the BIOS Flash Program Registers, PR0-PR4, GPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.

27.112 Host Flash Protected Range 1 (CSXE_HOST_PR1)—Offset 188h

These are not physical registers in the CSME memory space. These addresses provide the CSME with a read-only view of the values stored in the Host Protected Range registers.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.

27.113 Host Flash Protected Range 2 (CSXE_HOST_PR2)—Offset 18Ch

These are not physical registers in the CSME memory space. These addresses provide the CSME with a read-only view of the values stored in the Host Protected Range registers.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.

27.114 Host Flash Protected Range 3 (CSXE_HOST_PR3)—Offset 190h

These are not physical registers in the CSME memory space. These addresses provide the CSME with a read-only view of the values stored in the Host Protected Range registers.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.

27.115 Host Flash Protected Range 4 (CSXE_HOST_PR4)—Offset 194h

These are not physical registers in the CSME memory space. These addresses provide the CSME with a read-only view of the values stored in the Host Protected Range registers.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.

27.116 Host Global Protected Range 0 (CSXE_HOST_GPR0)—Offset 198h

These are not physical registers in the CSME memory space. These addresses provide the CSME with a read-only view of the values stored in the Host Protected Range registers.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the Host Flash Program Registers, PR0-PR4, GPR0.

27.117 IE Flash Protected Range 0 (CSXE_IE_PR0)—Offset 1B0h

These are not physical registers in the CSxE memory space. These addresses provide the CSxE with a read-only view of the values stored in the IE Protected Range registers.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.

27.118 IE Flash Protected Range 1 (CSXE_IE_PR1)—Offset 1B4h

These are not physical registers in the CSxE memory space. These addresses provide the CSxE with a read-only view of the values stored in the IE Protected Range registers.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.

27.119 IE Flash Protected Range 2 (CSXE_IE_PR2)—Offset 1B8h

These are not physical registers in the CSxE memory space. These addresses provide the CSxE with a read-only view of the values stored in the IE Protected Range registers.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.

27.120 IE Flash Protected Range 3 (CSXE_IE_PR3)—Offset 1BCh

These are not physical registers in the CSxE memory space. These addresses provide the CSxE with a read-only view of the values stored in the IE Protected Range registers.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.

27.121 IE Flash Protected Range 4 (CSXE_IE_PR4)—Offset 1C0h

These are not physical registers in the CSxE memory space. These addresses provide the CSxE with a read-only view of the values stored in the IE Protected Range registers.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the IE Flash Program Registers, PR0-PR4, GPR0.

27.122 BIOS Flash Primary Region (BIOS_BFPREG)—Offset 0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Shadowed BIOS Region Select (SBRS):** This bit reflects the CSXEFCTRL.BRS bit under the CSME SPI CSME root space configuration register.

Bit Range	Default & Access	Field Name (ID): Description
30:16	0h RO/V	**BIOS Flash Primary Region Limit (PRL):** This specifies address bits 26:12 for the Primary Region Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG1.Region Limit, or the Flash Descriptor.FLREG6.Region Limit depending on the BFPREG.SBRS bit.
15	0h RO	**Reserved (RSVD):** Reserved.
14:0	0h RO/V	**BIOS Flash Primary Region Base (PRB):** This specifies address bits 26:12 for the Primary Region Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG1.Region Base, or the Flash Descriptor.FLREG6.Region Base depending on the BFPREG.SBRS bit.

27.123 Hardware Sequencing Flash Status and Control (BIOS_HSFSTS_CTL)—Offset 4h

Several hardware sequenced operations are not supported in slave-attach flash mode.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 2000h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Flash SPI SMI# Enable (FSMIE):** When set to 1, the SPI asserts an SMI# request whenever the Flash Cycle Done bit is 1.
30	0h RO	**Reserved (RSVD):** Reserved.
29:24	0h RW	**Flash Data Byte Count (FDBC):** This field specifies the number of bytes to shift in or out during the data portion of the SPI cycle. The contents of this register are 0s based with 0b representing 1 byte and 3Fh representing 64 bytes. The number of bytes transferred is the value of this field plus 1. This field is ignored for the Block Erase command.
23:22	0h RW	**Reserved RW Scratch Pad (Reserved_SCRATCHPAD):** Scratch Pad bits that are R/W to be used during ECO.
21	0h RW	**Write Enable Type (WET):** 0: Use 06h as the write enable instruction 1: Use 50h as the write enable instruction Note: No supported flash devices require the 50h opcode to enable a non-volatile status register write. This bit is no longer required.

Bit Range	Default & Access	Field Name (ID): Description
20:17	0h RW	**Flash Cycle (FCYCLE):** This field defines the Flash SPI cycle type generated to the FLASH when the FGO bit is set as defined below: 0h: Read (1 up to 64 bytes by setting FDBC) 1h: Reserved 2h: Write (1 up to 64 bytes by setting FDBC) 3h: 4k Block Erase 4h: 64k Sector erase 5h: Read SFDP 6h: Read JEDEC ID 7h: write status 8h: read status 9h: RPMC Op1 (only legal in slave-attach flash mode) Ah: RPMC Op2 (only legal in slave-attach flash mode) Bh: Reserved Ch: Reserved Dh: Reserved Eh: Reserved Fh: Reserved Flash controller hardware automatically inserts a write enable opcode prior to Write and erase operations. Hardware automatically polls for device not-busy using read status after write and erase operations. If the device does not support 64k erase size (or if it doesn't support SFDP) then only 4k is allowed. See Program Register Access section for opcode values. Note: If reserved '1' is programmed to this field, flash controller will handle it as if it is 0 (Read) (Preserves legacy behavior.) Other reserved operations, including the RPMC, set the FCERR bit and do not issue a cycle on the SPI bus.
16	0h RW/1S/V	**Flash Cycle Go (FGO):** A write to this register with a '1' in this bit initiates a request to the Flash SPI Arbiter to start a cycle. This register is cleared by hardware when the cycle is granted by the SPI arbiter to run the cycle on the SPI bus. When the cycle is complete, the FDONE bit is set. Software is forbidden to write to any register in the HSFLCTL register between the FGO bit getting set and the FDONE bit being cleared. Any attempt to violate this rule will be ignored by hardware. Hardware allows other bits in this register to be programmed for the same transaction when writing this bit to 1. This saves an additional memory write. This bit remains set until the transaction is granted by the SPI Controller's internal arbiter.
15	0h RW/L	**Flash Configuration Lock-Down (FLOCKDN):** When set to 1, those Flash Program Registers that are locked down by this FLOCKDN bit cannot be written. Once set to 1, this bit can only be cleared by a hardware reset.

Bit Range	Default & Access	Field Name (ID): Description
14	0h RO/V	**Flash Descriptor Valid (FDV):** This bit is set to a 1 if the Flash Controller read the correct Flash Descriptor Signature. If the Flash Descriptor Valid bit is not 1, software cannot use the Hardware Sequencing registers, but must use the software sequencing registers. Any attempt to use the Hardware Sequencing registers will result in the FCERR bit being set.
13	1h RO/V	**Flash Descriptor Override Pin-Strap Status (FDOPSS):** This register indicates whether the flash controller is overriding descriptor permissions due to the Pin-Strap. Note: the register value is the inversion of the level sampled on the external pinstrap. 1: No override 0: The Flash Descriptor Overide strap is set
12	0h RW/L	**PRR3 PRR4 Lock-Down (PRR34_LOCKDN):** When set to 1, the BIOS PRR3 and PRR4 registers cannot be written. Once set to 1, this bit can only be cleared by a hardware reset. See also the BIOS PR3 and BIOS PR4 register descriptions. Note: This bit is not used if the DLOCK register is implemented.
11	0h RW/L	**Write Status Disable (WRSDIS):** 0: Write status operation may be issued using Hardware Sequencing. 1: Write status is not allowed as a Hardware Sequencing operation. The flash controller will block the operation and set the FCERR bit when software sets the 'go' bit.
10:9	0h RO	**Reserved (RSVD_1):** Reserved.
8	0h RW/1C/V	**SAF ctype error (H_SAF_CE):** Hardware sets this bit to 1 when a transaction is returned from the eSPI controller with ctype error.
7	0h RO/V	**SAF Mode Active (H_SAF_MODE_ACTIVE):** 0 : indicates flash is attached directly to the PCH via the SPI bus 1 : indicates flash is attached to the EC/BMC and is accessed via tunneled eSPI commands
6	0h RW/1C/V	**SAF link Error (H_SAF_LE):** Hardware sets this bit to 1 when a transaction is returned from the eSPI channel with link error.
5	0h RO/V	**SPI Cycle In Progress (H_SCIP):** Hardware sets this bit when software sets the Flash Cycle Go (FGO) bit in the Hardware Sequencing Flash Control register. This bit remains set until the cycle completes on the SPI interface or the transaction is prevented due to any protection policy violation (descriptor, address range, protected region, etc). Hardware automatically sets and clears this bit so that software can determine when read data is valid and/or when it is safe to begin programming the next command. Software must only program the next command when this bit is 0.
4	0h RW/1C/V	**SAF Data length Error (H_SAF_DLE):** Hardware sets this bit to 1 when a transaction is returned from the eSPI channel with an incorrect data length.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW/1C/V	**SAF Error (H_SAF_ERROR):** Hardware sets this bit to 1 when a transaction is requested that is not supported by slave-attached flash, e.g. read status.
2	0h RW/1C/V	**Access Error Log (H_AEL):** Hardware sets this bit to a 1 when an attempt was made to access the BIOS region using the direct access method or an access to the BIOS Program Registers that violated the security restrictions. This bit is simply a log of an access security violation. This bit is cleared by software writing a '1'.
1	0h RW/1C/V	**Flash Cycle Error (FCERR):** Hardware sets this bit to 1 when a program register access is blocked to the FLASH due to one of the protection policies or when any of the programmed cycle registers is written while a programmed access is already in progress. This bit remains asserted until cleared by software writing a 1 or until a partition reset occurs. Software must clear this bit before setting the FLASH Cycle GO bit in this register.
0	0h RW/1C/V	**Flash Cycle Done (FDONE):** The PCH sets this bit to 1 when the SPI Cycle completes after software previously set the FGO bit. This bit remains asserted until cleared by software writing a 1 or partition reset. When this bit is set and the SPI SMI Enable bit is set the flash controller sends SMI messages. Software must make sure this bit is cleared prior to enabling the SPI SMI assertion for a new programmed access.

27.124 Flash Address (BIOS_FADDR)—Offset 8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:27	0h RO	**Reserved (RSVD):** Reserved.
26:0	0h RW	**Flash Linear Address (FLA):** The FLA is the starting byte linear address of a SPI Read or Write cycle or an address within a Block for the Block Erase command. The Flash Linear Address must fall within a region for which BIOS has access permissions. Hardware must convert the FLA into a Flash Physical Address (FPA) before running this cycle on the SPI bus.

27.125 Discrete Lock Bits (BIOS_DLOCK)—Offset Ch

Lockable BIOS registers may be locked by either the global FLOCKDN bit or by the individual DLOCK.* bit. Each lockable bit in this register is locked either by itself or by the FLOCKDN bit.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RO	**Reserved (RSVD):** Reserved.
16	0h RW/L	**SSEQ Lock-Down (SSEQLOCKDN):** BIOS Software Sequencing registers are locked when the logical OR of this bit and FLOCKDN is true. The affected registers are SSFSTS_CTL.SCF, PREOP_OPTYPE, OPMENU0, and OPMENU1.**Once set to 1 this register is only cleared by host partition reset.**

27.126 Flash Data 0 (BIOS_FDATA0)—Offset 10h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 0 (FD0):** This field is shifted out as the SPI Data on the Master-Out Slave-In Data pin during the data portion of the SPI cycle. This register also shifts in the data from the Master-In Slave-Out pin into this register during the data portion of the SPI cycle. The data is always shifted starting with the least significant byte, msb to lsb, followed by the next least significant byte, msb to lsb, etc. Specifically, the shift order on SPI in terms of bits within this register is: 7-6-5-4-3-2-1-0-15-14-13- ... 8-23-22- ... 16-31 ... 24 Bit 24 is the last bit shifted out/in. There are no alignment assumptions; byte 0 always represents the value specified by the cycle address. Note that the data in this register may be modified by the hardware during any programmed SPI transaction. Direct Memory Reads do not modify the contents of this register.

27.127 Flash Data 1 (BIOS_FDATA1)—Offset 14h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 1 (FD1):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD0 has completely shifted in/out.

27.128 Flash Data 2 (BIOS_FDATA2)—Offset 18h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 2 (FD2):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD1 has completely shifted in/out.

27.129 Flash Data 3 (BIOS_FDATA3)—Offset 1Ch

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 3 (FD3):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD2 has completely shifted in/out.

27.130 Flash Data 4 (BIOS_FDATA4)—Offset 20h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 4 (FD4):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD3 has completely shifted in/out.

27.131 Flash Data 5 (BIOS_FDATA5)—Offset 24h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 5 (FD5):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD4 has completely shifted in/out.

27.132 Flash Data 6 (BIOS_FDATA6)—Offset 28h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 6 (FD6):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD5 has completely shifted in/out.

27.133 Flash Data 7 (BIOS_FDATA7)—Offset 2Ch

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 7 (FD7):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD6 has completely shifted in/out.

27.134 Flash Data 8 (BIOS_FDATA8)—Offset 30h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 8 (FD8):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD7 has completely shifted in/out.

27.135 Flash Data 9 (BIOS_FDATA9)—Offset 34h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 9 (FD9):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD8 has completely shifted in/out.

27.136 Flash Data 10 (BIOS_FDATA10)—Offset 38h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 10 (FD10):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD9 has completely shifted in/out.

27.137 Flash Data 11 (BIOS_FDATA11)—Offset 3Ch

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 11 (FD11):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD10 has completely shifted in/out.

27.138 Flash Data 12 (BIOS_FDATA12)—Offset 40h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 12 (FD12):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD11 has completely shifted in/out.

27.139 Flash Data 13 (BIOS_FDATA13)—Offset 44h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 13 (FD13):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD12 has completely shifted in/out.

27.140 Flash Data 14 (BIOS_FDATA14)—Offset 48h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 14 (FD14):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD13 has completely shifted in/out.

27.141 Flash Data 15 (BIOS_FDATA15)—Offset 4Ch

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**Flash Data 15 (FD15):** Similar definition as Flash Data 0. However, this register does not begin shifting until FD14 has completely shifted in/out.

27.142 Flash Region Access Permissions (BIOS_FRACC)—Offset 50h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:42C2h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW/L	**BIOS Master Write Access Grant (BMWAG):** Each bit [31:24] corresponds to Master[7:0]. BIOS can grant one or more masters write access to the BIOS region 1 overriding the permissions in the Flash Descriptor.
23:16	0h RW/L	**BIOS Master Read Access Grant (BMRAG):** Each bit [23:16] corresponds to Master[7:0]. BIOS can grant one or more masters read access to the BIOS region 1 overriding the read permissions in the Flash Descriptor.
15:8	42h RO/V	**BIOS Region Write Access (BRWA):** Each bit [15:8] corresponds to Regions [7:0]. If the bit is set, this master can erase and write that particular region through register accesses. The contents of this register are that of the Flash Descriptor.Flash Master 1.Master Region Write Access OR a particular master has granted BIOS write permissions in their Master Write Access Grant register OR the Flash Descriptor Security Override strap is set. BIOS always have the write access to its own Region 1 and Region 6 by default. Thus, the reset default of this field will be read as '42h'.

Bit Range	Default & Access	Field Name (ID): Description
7:0	C2h RO/V	**BIOS Region Read Access (BRRA):** Each bit [7:0] corresponds to Regions [7:0]. If the bit is set, this master can read that particular region through register accesses. The contents of this register are that of the Flash Descriptor.Flash Master 1.Master Region Read Access OR a particular master has granted BIOS read permissions in their Master Read Access Grant register OR the Flash Descriptor Security Override strap is set. BIOS always have the read access to its own Regions 1, 6, and 7 by default. Thus, the reset default of this field will be read as 'C2h'.

27.143 Flash Region 0 (BIOS_FREG0)—Offset 54h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 0 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG0.Region Limit.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	0h RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 0 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG0.Region Base.

27.144 Flash Region 1 (BIOS_FREG1)—Offset 58h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 1 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG1.Region Limit.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 1 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG1.Region Base.

27.145 Flash Region 2 (BIOS_FREG2)—Offset 5Ch

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 2 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG2.Region Limit.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 2 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG2.Region Base.

27.146 Flash Region 3 (BIOS_FREG3)—Offset 60h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 3 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG3.Region Limit.
15	0h RO	**Reserved (RSVD_1):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 3 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG3.Region Base.

27.147 Flash Region 4 (BIOS_FREG4)—Offset 64h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 4 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG4.Region Limit.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 4 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG4.Region Base.

27.148 Flash Region 5 (BIOS_FREG5)—Offset 68h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 5 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG5.Region Limit.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 5 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG5.Region Base.

27.149 Flash Region 6 (BIOS_FREG6)—Offset 6Ch

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 6 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG6.Region Limit if the Server Feature Disable Fuse 6 is '0'. If the Server Feature Disable Fuse 6 is a '1' then this field is loaded with all zeros.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 6 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG6.Region Base if the Server Feature Disable Fuse 6 is '0'. If the Server Feature Disable Fuse 6 is a '1' then this field is loaded with all ones.

27.150 Flash Region 7 (BIOS_FREG7)—Offset 70h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 7 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG7.Region Limit.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 7 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG7.Region Base.

27.151 Flash Region 8 (BIOS_FREG8)—Offset 74h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 8 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG8.Region Limit.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 8 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG8.Region Base.

27.152 Flash Region 9 (BIOS_FREG9)—Offset 78h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved.
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 9 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG9.Region Limit.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 9 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG9.Region Base.

27.153 Flash Region 10 (BIOS_FREG10)—Offset 7Ch

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 10 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG10.Region Limit
15	0h RO	**Reserved (RSVD_1):** Reserved
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 10 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG10.Region Base

27.154 Flash Region 11 (BIOS_FREG11)—Offset 80h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 11 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG11.Region Limit
15	0h RO	**Reserved (RSVD_1):** Reserved
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 11 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG11.Region Base

27.155 Flash Protected Range 0 (BIOS_FPR0)—Offset 84h

This register cannot be written when the FLOCKDN bit is set to 1.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/L	**Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
30:16	0h RW/L	**Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range. Address bits 11:0 are assumed to be FFFh for the limit comparison. Any address greater than the value programmed in this field is unaffected by this protected range.
15	0h RW/L	**Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that reads directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
14:0	0h RW/L	**Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range. Address bits 11:0 are assumed to be 000h for the base comparison. Any address less than the value programmed in this field is unaffected by this protected range.

27.156 Flash Protected Range 1 (BIOS_FPR1)—Offset 88h

This register can not be written when the FLOCKDN bit is set to 1.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/L	**Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
30:16	0h RW/L	**Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range. Address bits 11:0 are assumed to be FFFh for the limit comparison. Any address greater than the value programmed in this field is unaffected by this protected range.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/L	**Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that reads directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
14:0	0h RW/L	**Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range. Address bits 11:0 are assumed to be 000h for the base comparison. Any address less than the value programmed in this field is unaffected by this protected range.

27.157 Flash Protected Range 2 (BIOS_FPR2)—Offset 8Ch

This register can not be written when the FLOCKDN bit is set to 1.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/L	**Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
30:16	0h RW/L	**Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range. Address bits 11:0 are assumed to be FFFh for the limit comparison. Any address greater than the value programmed in this field is unaffected by this protected range.
15	0h RW/L	**Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that reads directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
14:0	0h RW/L	**Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range. Address bits 11:0 are assumed to be 000h for the base comparison. Any address less than the value programmed in this field is unaffected by this protected range.

27.158 Flash Protected Range 3 (BIOS_FPR3)—Offset 90h

If the DLOCK register is not implemented, then this register cannot be written when the PRR34_LOCKDN bit is set to 1. Else this register cannot be written when either FLOCKDN or the DLOCK.PRnLOCKDN corresponding to this register is set.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/L	**Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
30:16	0h RW/L	**Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range. Address bits 11:0 are assumed to be FFFh for the limit comparison. Any address greater than the value programmed in this field is unaffected by this protected range.
15	0h RW/L	**Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that reads directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
14:0	0h RW/L	**Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range. Address bits 11:0 are assumed to be 000h for the base comparison. Any address less than the value programmed in this field is unaffected by this protected range.

27.159 Flash Protected Range 4 (BIOS_FPR4)—Offset 94h

If the DLOCK register is not implemented, then this register cannot be written when the PRR34_LOCKDN bit is set to 1. Else this register cannot be written when either FLOCKDN or the DLOCK.PRnLOCKDN corresponding to this register is set.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/L	**Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
30:16	0h RW/L	**Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range. Address bits 11:0 are assumed to be FFFh for the limit comparison. Any address greater than the value programmed in this field is unaffected by this protected range.
15	0h RW/L	**Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that reads directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
14:0	0h RW/L	**Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range. Address bits 11:0 are assumed to be 000h for the base comparison. Any address less than the value programmed in this field is unaffected by this protected range.

27.160 Global Protected Range 0 (BIOS_GPR0)—Offset 98h

This register is initialized via softstraps. This protected range applies globally to all masters / flash requesters. If SPT users do not want a register that applies across all masters then the enable bits must be set false. Since this register is a descriptor-based protection, it is disabled (both enable bits are treated as false) when the Flash Descriptor Security Override pinstrap is asserted. Note: since this register is a RO view of the softstraps the underlying values are only reset when softstraps are reset.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.

Bit Range	Default & Access	Field Name (ID): Description
30:16	0h RO/V	**Protected Range Limit (PRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range. Address bits 11:0 are assumed to be FFFh for the limit comparison. Any address greater than the value programmed in this field is unaffected by this protected range. Note: If either Write or Read protection is enabled, then Limit must be configured greater than or equal to Base.
15	0h RO/V	**Read Protection Enable (RPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that reads directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
14:0	0h RO/V	**Protected Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range. Address bits 11:0 are assumed to be 000h for the base comparison. Any address less than the value programmed in this field is unaffected by this protected range. Note: If either Write or Read protection is enabled, then Limit must be configured greater than or equal to Base.

27.161 Software Sequencing Flash Status and Control (BIOS_SSFSTS_CTL)—Offset A0h

The Software Sequencing control and status registers are intended to be used only as a back-up mode to the hardware sequencing control and status registers.**When HSSEN = 0 the SCGO bit must become read-only.** It is an implementation decision whether to make all register bits RO when HSSEN=0. Software sequencing is not supported in slave-attach flash mode.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:FE000000h

Bit Range	Default & Access	Field Name (ID): Description
31:27	1Fh RO	**Reserved (RSVD):** Reserved: This field is not programmable, it is fixed at 1Fh.

Bit Range	Default & Access	Field Name (ID): Description
26:24	6h RW/L	**SPI Cycle Frequency (SCF):** The listed frequencies are approximate. 000: 120MHz (not supported in SPT) 001: 60MHz (not supported in SPT) 010: 48MHz 011: 40MHz (not supported in SPT) 100: 30MHz 101: 24MHz (not supported in SPT) 110: 17MHz All others: Reserved This register sets frequency to use for all SPI Software Sequencing cycles (write, erase, fast read, read status, etc) except for the Read cycle which always run at 20MHz. This register is locked when the SPI Configuration Lock-Down (FLOCKDN) bit is set. Note: Fast Reset Testmode overrides the value programmed into this register.
23	0h RW	**SPI SMI Enable (SME):** When set to 1, the SPI sends an assert SMI when the Cycle Done Status bit changes to a 1.
22	0h RW	**Data Cycle (DS):** When set to 1, there is data that corresponds to this transaction. When 0, no data is delivered for this cycle, and the DBC and data fields themselves are do not care.
21:16	0h RW	**Data Byte Count (DBC):** This field specifies the number of bytes to shift in or out during the data portion of the SPI cycle. The valid settings (in decimal) are any value from 0 to 63. The number of bytes transferred is the value of this field plus 1. Note that when this field is 00_0000b, then there is 1 byte to transfer and that 11_1111b means there are 64 bytes to transfer.
15	0h RO	**Reserved (RSVD_1):** Reserved.
14:12	0h RW	**Cycle Opcode Pointer (COP):** This field selects one of the programmed opcodes in the Opcode Menu to be used as the SPI Command/Opcode. In the case of an Atomic Cycle Sequence, this determines the second command.
11	0h RW	**Sequence Prefix Opcode Pointer (SPOP):** This field selects one of the two programmed prefix opcodes for use when performing an Atomic Cycle Sequence. A value of 0 points to the opcode in the least significant byte of the Prefix Opcodes register. By making this programmable, the ICH supports flash devices that have different opcodes for enabling writes to the data space vs. status register.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW	**Atomic Cycle Sequence (ACS):** When set to 1 along with the SCGO assertion, the PCH will execute a sequence of commands on the SPI interface without allowing another master to arbitrate and interleave cycles. The sequence is composed of: Atomic Sequence Prefix Command (8-bit opcode only) Primary Command specified below by software (can include address and data) Polling the Flash Status Register (opcode 05h) until bit 0 becomes 0b. The SPI Cycle in Progress bit remains set and the Cycle Done Status bit remains unset until the Busy bit in the Flash Status Register returns 0.
9	0h RW/1S/V	**SPI Cycle Go (SCGO):** A write to this register with a '1' in this bit starts the SPI cycle defined by the other bits of this register. The SPI Cycle in Progress (SCIP) bit gets set by this action. Hardware must ignore writes to this bit while the Cycle In Progress bit is set. Hardware allows other bits in this register to be programmed for the same transaction when writing this bit to 1. This saves an additional memory write. This bit remains set until the transaction is granted by the SPI Controller's internal arbiter.
8	0h RO	**Reserved (RSVD_2):** Reserved.
7	0h RO/V	**Fast Read Supported (FAST_READ_SUPPORT):** This bit reflects the value of the Fast Read Support bit in the Flash Descriptor Component Section.
6	0h RO/V	**Dual Output Fast Read Supported (DUAL_FAST_READ_SUPPORT):** This bit reflects the value of the Dual Output Fast Read Support bit in the Flash Descriptor Component Section.
5	0h RO	**Reserved (RSVD_3):** Reserved.
4	0h RO/V	**Access Error Log (AEL):** This bit reflects the value of the Hardware Sequencing Status.AEL register.
3	0h RW/1C/V	**Flash Cycle Error (FCERR):** Hardware sets this bit to 1 when a programmed access is blocked from running on the SPI interface due to one of the protection policies or when any of the programmed cycle registers is written while a programmed access is already in progress. This bit remains asserted until cleared by software writing a 1 or partition reset.
2	0h RW/1C/V	**Cycle Done Status (CYCLE_DONE_STS):** The PCH sets this bit to 1 when the SPI Cycle completes (i.e., SCIP bit is 0) after software sets the GO bit. This bit remains asserted until cleared by software writing a 1 or hardware reset. When this bit is set and the SPI SMI# Enable bit is set, an internal signal is asserted to the SMI# generation block. Software must make sure this bit is cleared prior to enabling the SPI SMI# assertion for a new programmed access.
1	0h RO	**Reserved (RSVD_4):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO/V	**SPI Cycle In Progress (SCIP):** Hardware sets this bit when software sets the SPI Cycle Go bit in the Command register. This bit remains set until the cycle completes on the SPI interface or the transaction is prevented due to any protection policy violation (descriptor, address range, protected region, etc). Hardware automatically sets and clears this bit so that software can determine when read data is valid and/or when it is safe to begin programming the next command. Software must only program the next command when this bit is 0.

27.162 Prefix Opcode and Opcode Type Configuration (BIOS_PREOP_OPTYPE)—Offset A4h

This register is not writable when the Flash Configuration Lock-Down (FLOCKDN) bit is set. Entries in this register correspond to the entries in the Opcode Menu Configuration register. Note that the definition below only provides write protection for opcodes that have addresses associated with them. Therefore, any erase or write opcodes that do not use an address should be avoided (for example, 'Chip Erase' and 'Auto-Address Increment Byte Program').

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RW/L	**Opcode Type 7 (OPCODE_TYPE7):** See the description for bits 1:0.
29:28	0h RW/L	**Opcode Type 6 (OPCODE_TYPE6):** See the description for bits 1:0.
27:26	0h RW/L	**Opcode Type 5 (OPCODE_TYPE5):** See the description for bits 1:0.
25:24	0h RW/L	**Opcode Type 4 (OPCODE_TYPE4):** See the description for bits 1:0.
23:22	0h RW/L	**Opcode Type 3 (OPCODE_TYPE3):** See the description for bits 1:0.
21:20	0h RW/L	**Opcode Type 2 (OPCODE_TYPE2):** See the description for bits 1:0.
19:18	0h RW/L	**Opcode Type 1 (OPCODE_TYPE1):** See the description for bits 1:0.

Bit Range	Default & Access	Field Name (ID): Description
17:16	0h RW/L	**Opcode Type 0 (OPCODE_TYPE0):** This field specifies information about the corresponding Opcode 0. This information allows the hardware to 1) know whether to use the address field and 2) provide BIOS and Shared Flash protection capabilities. The hardware implementation also uses the read vs. write information for modifying the behavior of the SPI interface logic. The encoding of the two bits is: 00: No Address associated with this Opcode and Read Cycle type 01: No Address associated with this Opcode and Write Cycle type 10: Address required; Read cycle type 11: Address required; Write cycle type
15:8	0h RW/L	**Prefix Opcode 1 (PREFIX_OPCODE1):** Software programs an SPI opcode into this field that is permitted to run as the first command in an atomic cycle sequence.
7:0	0h RW/L	**Prefix Opcode 0 (PREFIX_OPCODE0):** Software programs an SPI opcode into this field that is permitted to run as the first command in an atomic cycle sequence.

27.163 Opcode Menu0 Configuration (BIOS_OPMENU0)—Offset A8h

This register is not writable when the SPI Configuration Lock-Down bit (FLOCKDN) is set. Eight entries are available in this register to give BIOS a sufficient set of commands for communicating with the flash device, while also restricting what malicious software can do. This keeps the hardware flexible enough to operate with a wide variety of SPI devices. It is recommended that BIOS avoid programming Write Enable opcodes in this menu. Malicious software could then perform writes and erases to the SPI flash without using the atomic cycle mechanism. This could cause functional failures in a shared flash environment. Write Enable opcodes should only be programmed in the Prefix Opcodes.
Eight entries are available in this register to give BIOS a sufficient set of commands for communicating with the flash device, while also restricting what malicious software can do. This keeps the hardware flexible enough to operate with a wide variety of SPI devices.
It is recommended that BIOS avoid programming Write Enable opcodes in this menu. Malicious software could then perform writes and erases to the SPI flash without using the atomic cycle mechanism. This could cause functional failures in a shared flash environment. Write Enable opcodes should only be programmed in the Prefix Opcodes.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW/L	**Allowable Opcode 3 (OPCODE3):** See the description for bits 7:0.

Bit Range	Default & Access	Field Name (ID): Description
23:16	0h RW/L	**Allowable Opcode 2 (OPCODE2):** See the description for bits 7:0.
15:8	0h RW/L	**Allowable Opcode 1 (OPCODE1):** See the description for bits 7:0.
7:0	0h RW/L	**Allowable Opcode 0 (OPCODE0):** Software programs an SPI opcode into this field for use when initiating SPI commands through the Control Register.

27.164 Opcode Menu1 Configuration (BIOS_OPMENU1)—Offset ACh

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW/L	**Allowable Opcode 7 (OPCODE7):** See the description in OPMENU0.
23:16	0h RW/L	**Allowable Opcode 6 (OPCODE6):** See the description in OPMENU0.
15:8	0h RW/L	**Allowable Opcode 5 (OPCODE5):** See the description in OPMENU0.
7:0	0h RW/L	**Allowable Opcode 4 (OPCODE4):** See the description in OPMENU0.

27.165 Secondary Flash Region Access Permissions (BIOS_SFRACC)—Offset B0h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW/L	**Secondary BIOS Master Write Access Grant (SECONDARYBIOS_MWAG):** Each bit [31:29] corresponds to Master[7:0]. BIOS can grant one or more masters write access to the Secondary BIOS region 6 overriding the permissions in the Flash Descriptor.

Bit Range	Default & Access	Field Name (ID): Description
23:16	0h RW/L	**Secondary BIOS Master Read Access Grant (SECONDARYBIOS_MRAG):** Each bit [28:16] corresponds to Master[7:0]. BIOS can grant one or more masters read access to the Secondary BIOS region 6 overriding the read permissions in the Flash Descriptor.
15:0	0h RO	**Reserved (RSVD):** Reserved.

27.166 Flash Descriptor Observability Control (BIOS_FDOC)—Offset B4h

This is a test mode only register that can be used to observe the contents of the Flash Descriptor that is stored internally in the PCH Flash Controller. The CPU Complex soft straps are not observable in PCH as these are forwarded to the CPU Complex and not stored.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD):** Reserved.
14:12	0h RW	**Flash Descriptor Section Select (FDSS):** Selects which section within the loaded Flash Descriptor to observe. 000 : Flash Signature and Descriptor Map 001 : Component 010 : Region 011 : Master 100 : PCH Soft Straps Other : Reserved
11:2	0h RW	**Flash Descriptor Section Index (FDSI):** Selects the DW offset within the Flash Descriptor Section to observe.
1:0	0h RO	**Reserved (RSVD_1):** Reserved.

27.167 Flash Descriptor Observability Data (BIOS_FDOD)—Offset B8h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Flash Descriptor Section Data (FDSD):** Returns the DW of data to observe as selected in the Flash Descriptor Observability Control.

27.168 Additional Flash Control (BIOS_AFC)—Offset C0h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RO	**Reserved (RSVD):** Reserved.
9:1	0h RW	**Reserved RW Scratch Pad (Reserved_SCRATCHPAD):** Scratch Pad bits that are R/W to be used during ECO.
0	0h RW/V/P	**Stop Prefetch on Flush Pending (SPFP):** When set to '1', the in progress of a prefetch will be ended if subsequence access from the master of the same Root Space is detected to be a cache-miss and read cache will be flushed. When set to '0', the prefetch will be allowed to complete prior to flushing. This register is only reset on an Early boot reset of both the Host and the CSME partitions. Hardware loads this bit from the PCH soft strap SPI_SPFP when the strap completion is received from the strap pull message.

27.169 Vendor Specific Component Capabilities for Component 0 (BIOS_SFDP0_VSCC0)—Offset C4h

Support for non-SFDP capable flash devices is deprecated.
Hardware uses the data it loads from the flash device's SFDP tables. Hardware copies some SFDP data to the SFDP_VSCCn registers for the convenience of BIOS/FW, but does not directly use the data in those registers. Hardware locks the SFDP_VSCCn registers after copying SFDP data into them and setting the CPPTV bit.
Firmware requirements:
The QER bits are not used by the hardware. The QER bits are used by software/ firmware to ensure the QE bit is not changed while doing status register write. The QER field in SFDP_VSCC is only valid if the flash device contains a rev 1.5 or higher SFDP table. Firmware must use PTINX/PTDATA to read the Parameter Table Header to determine the SFDP revision.
If the SFDP revision is < 1.5 then firmware must use some other means to determine QER, e.g. the Flash Upper Map VSCC tables.
The fields in this register pertain to cycles targeting addresses within Component 0.
The lockable bits in this register are locked when either CPPTV is set to '1' by hardware or when VCL is '1'.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:2000h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Component Property Parameter Table Valid (CPPTV):** This bit is set to a 1 if the Flash Controller detects a valid SFDP Component Property Parameter Table in Component 0. Note: If this bit is set software must not overwrite bits that were initialized by hardware via SFDP discovery.
30	0h RW/L	**Vendor Component Lock (VCL):** 0: The lock bit is not set. 1: The Vendor Component Lock bit is set. This register locks itself when set.
29	0h RW/V/L	**64k Erase Valid (EO_64k_VALID):** 0: The EO_64k opcode is not valid. 1: The EO_64k opcode is valid.
28	0h RW/V/L	**4k Erase Valid (EO_4k_VALID):** 0: The EO_4k opcode is not valid. 1: The EO_4k opcode is valid.
27	0h RW/L	**RPMC Supported (RPMC_SUPPORTED):** 0: The device does not support RPMC. 1: The device supports RPMC.
26	0h RW/V/L	**Deep Powerdown Supported (DEEP_PWRDN_SUPPORTED):** 0: The device does not support Deep Powerdown. 1: The device supports Deep Powerdown.
25	0h RW/V/L	**Suspend/Resume Supported (SUSPEND_RESUME_SUPPORTED):** 1: The device does not support Suspend/Resume. 0: The device supports Suspend/Resume.
24	0h RW/V/L	**Soft Reset Supported (SOFT_RST_SUPPORTED):** 0: The device does not support Soft Reset. 1: The device supports Soft Reset.
23:16	0h RW/V/L	**64k Erase Opcode (EO_64k):** This register is programmed with the Flash 64k sector erase instruction opcode for component 0. This register is locked by the Vendor Component Lock (VCL) bit or the CPPTV bit.
15:8	20h RW/V/L	**4k Erase Opcode (EO_4k):** This register is programmed with the Flash 4k subsector erase instruction opcode for component 0. Software must program this register if the SFDP table for this component does not show 4 kByte erase capability. This register is locked by the Vendor Component Lock (VCL) bit or the CPPTV bit.

Bit Range	Default & Access	Field Name (ID): Description
7:5	0h RW/V/L	**Quad Enable Requirements (QER):** This field is defined by the JEDEC JESD216A standard for SFDP, reference 2. Information is copied here for reference. The JEDEC standard shall take precedence in the event of a discrepancy. This field describes whether the device contains a Quad Enable (QE) bit used to enable 1-1-4 and 1-4-4 quad read or quad program operations. If QE exists, this field also identifies the bit location and method to set/clear the bit. In this specification, status register 1 refers to the first data byte transferred on a Read Status (05h) or Write Status (01h) command. Status register 2 refers to the byte read using instruction 35h. Status register 2 is the second byte transferred in a Write Status (01h) command. Bits are numbered from 7 to 0, where bit 7 is transferred first on the wire. Note: Industry naming and definitions of these status registers may differ. The user will typically perform a read-modify-write sequence of operations to maintain the state of all other writable status register bits. For example read both status registers, set/clear QE, Write Status with both data bytes. 000b: Device does not have a QE bit. Device detects 1-1-4 and 1-4-4 reads based on instruction. DQ3/HOLD# functions as hold during instruction phase. 001b: QE is bit 1 of status register 2. It is set via Write Status with two data bytes where bit 1 of the second byte is one. It is cleared via Write Status with two data bytes where bit 1 of the second byte is zero. Writing only one byte to the status register has the side-effect of clearing status register 2, including the QE bit. The 100b code is used if writing one byte to the status register does not modify status register 2. 010b: QE is bit 6 of status register 1. It is set via Write Status with one data byte where bit 6 is one. It is cleared via Write Status with one data byte where bit 6 is zero. 011b: QE is bit 7 of status register 2. It is set via Write status register 2 instruction 3Eh with one data byte where bit 7 is one. It is cleared via Write status register 2 instruction 3Eh with one data byte where bit 7 is zero. The status register 2 is read using instruction 3Fh. 100b: QE is bit 1 of status register 2. It is set via Write Status with two data bytes where bit 1 of the second byte is one. It is cleared via Write Status with two data bytes where bit 1 of the second byte is zero. In contrast to the 001b code, writing one byte to the status register does not modify status register 2. 101b: QE is bit 1 of the status register 2. Status register 1 is read using Read Status instruction 05h. Status register 2 is read using instruction 35h. QE is set via Write Status instruction 01h with two data bytes where bit 1 of the second byte is one. It is cleared via Write Status with two data bytes where bit 1 of the second byte is zero. Other: Reserved. This register is locked by the Vendor Component Lock (VCL) bit. The flash controller hardware does not use this field. NOTE: This field is only valid for SFDP revision 1.5 and above. Firmware must check the revision prior to acting on the contents of this register.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW/V/L	**Write Enable on Write Status (WEWS):** 0: 50h is the opcode to enable a status register write. 1: 06h is the opcode to enable a status register write. This register is locked by the Vendor Component Lock (VCL) bit or the CPPTV bit. Note: Hardware ignores the state of this bit and only issues 06h as a write enable. See the specification for a Lynxpoint generation SPI controller for a description of how this bit was previously used.
3	0h RW/V/L	**Write Status Required (WSR):** 0: No requirement to write to the Write Status Register prior to a write. 1: A write is required to the Write Status Register prior to write and erase to remove any protection. This is required for SST components. This register is locked by the Vendor Component Lock (VCL) bit or the CPPTV bit. Note: Hardware ignores the state of this bit and behaves as if WSR=0. Flash devices that contain a 1 in this field are no longer supported by the flash controller. See the specification for a Lynxpoint generation SPI controller for a description of how this bit was previously used.
2	0h RW/V/L	**Write Granularity (WG):** 0 : Reserved. 1 : 64 Byte. This register is locked by the Vendor Component Lock (VCL) bit or the CPPTV bit. Note: Hardware ignores the state of this bit.
1:0	0h RW/L	**Reserved (RSVD):** Reserved.

27.170 Vendor Specific Component Capabilities for Component 1 (BIOS_SFDP1_VSCC1)—Offset C8h

The fields in this register pertain to cycles targeting addresses outside of Component 0. The lockable bits in this register are locked when either CPPTV is set to '1' by hardware or when VCL is '1'.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:2000h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Component Property Parameter Table Valid (CPPTV):** This bit is set to a 1 if the Flash Controller detects a valid SFDP Component Property Parameter Table in Component 1. Note: If this bit is set software must not overwrite bits that were initialized by hardware via SFDP discovery.

Bit Range	Default & Access	Field Name (ID): Description
30	0h RO	**Reserved (RSVD):** Reserved.
29	0h RW/V/L	**64k Erase Valid (EO_64k_VALID):** 0 The EO_64k opcode is not valid. 1 The EO_64k opcode is valid.
28	0h RW/V/L	**4k Erase Valid (EO_4k_VALID):** 0 The EO_4k opcode is not valid. 1 The EO_4k opcode is valid.
27	0h RW/L	**RPMC Supported (RPMC_SUPPORTED):** 0 The device does not support RPMC. 1 The device supports RPMC.
26	0h RW/V/L	**Deep Powerdown Supported (DEEP_PWRDN_SUPPORTED):** 0 The device does not support Deep Powerdown. 1 The device supports Deep Powerdown.
25	0h RW/V/L	**Suspend/Resume Supported (SUSPEND_RESUME_SUPPORTED):** 1: The device does not support Suspend/Resume. 0: The device supports Suspend/Resume.
24	0h RW/V/L	**Soft Reset Supported (SOFT_RST_SUPPORTED):** 0: The device does not support Soft Reset. 1: The device supports Soft Reset.
23:16	0h RW/V/L	**64k Erase Opcode (EO_64k):** This register is programmed with the Flash 64k sector erase instruction opcode for component 1. This register is locked by the Vendor Component Lock (VCL) bit or the CPPTV bit.
15:8	20h RW/V/L	**4k Erase Opcode (EO_4k):** This register is programmed with the Flash 4k subsector erase instruction opcode for component 1. Software must program this register if the SFDP table for this component does not show 4 kByte erase capability. This register is locked by the Vendor Component Lock (VCL) bit or the CPPTV bit.
7:5	0h RW/V/L	**Quad Enable Requirements (QER):** See description in BIOS_SFDP0_VSCC0.QER field.
4	0h RW/V/L	**Write Enable on Write Status (WEWS):** See description in BIOS_SFDP0_VSCC0.WEWS field.
3	0h RW/V/L	**Write Status Required (WSR):** See description in BIOS_SFDP0_VSCC0.WSR field.
2	0h RW/V/L	**Write Granularity (WG):** See description in BIOS_SFDP0_VSCC0.WG field.
1:0	0h RW/L	**Reserved (RSVD_1):** Reserved.

27.171 Parameter Table Index (BIOS_PTINX)—Offset CCh

Observability control for Component Property Tables. Note: The PTINX and PTDATA registers do not have any meaning in slave-attach flash mode because the SPI controller does not perform SFDP discovery.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved.
15:14	0h RW	**Supported Parameter Table (SPT):** Selects which supported parameter table to observe. 00: Component 0 Property Parameter Table. 01: Component 1 Property Parameter Table. 10 - 11: Reserved.
13:12	0h RW	**Header or Data (HORD):** Select parameter table header DW vs Data DW. 00: SFDP Header. 01: Parameter Table Header. 10: Data. 11: Reserved.
11:2	0h RW	**Parameter Table DW Index (PTDWI):** Selects the DW offset within the parameter table to observe. The returned data is undefined if the index is programmed to a value greater than the size of the header or table.
1:0	0h RO	**Reserved (RSVD_1):** Reserved.

27.172 Parameter Table Data (BIOS_PTDATA)—Offset D0h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**Parameter Table DW Data (PTDWD):** Returns the DW of data to observe as selected in the Parameter Table Index register. Note: The returned value of reserved fields in the SFDP header or table must be ignored by software. The flash controller may return either 0 or 1 for these fields.

27.173 SPI Bus Requester Status (BIOS_SBRS)—Offset D4h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**TPM Access Ongoing (TPM_ACC_ONG)**
30	0h RO/V	**eSPI Access Ongoing (ESPI_ACC_ONG):** This bit is only defined if eSPI and SPI are sharing the SPI bus.
29	0h RO/V	**Touch access ongoing (TOUCH_ACC_ONG):** <product specific = [quote]BXT-P, CNP, KBP[/quote]>
28	0h RO/V	**CSME accessing the Huffman Decompression (CSME_ACC_HWD):** <product specific = [quote]CNP, KBP[/quote]>
27:18	0h RO	**Reserved (RSVD):** Reserved
17:15	0h RO/V	**Master 5 Status (M5STATUS):** See description under M1STATUS. Note: Master numbering was changed in a late revision of the Cspec, after this register was already implemented in RTL. Hence the master numbering in this register is not consecutive.
14:12	0h RO/V	**Master 6 Status (M6STATUS):** See description under M1STATUS.
11:9	0h RO/V	**Master 4 Status (M4STATUS):** See description under M1STATUS.
8:6	0h RO/V	**Master 3 Status (M3STATUS):** See description under M1STATUS.
5:3	0h RO/V	**Master 2 Status (M2STATUS):** See description under M1STATUS.
2:0	0h RO/V	**Master 1 Status (M1STATUS):** Indicates whether this master has an outstanding transaction enqueued or in flight and the transaction type. 0xx: No transaction. 100: Flash read transaction. 101: Flash write transaction. 110: Flash erase transaction. 111: Flash RPMC transaction.

27.174 Flash Region 12 (BIOS_FREG12)—Offset E0h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 12 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG12.Region Limit
15	0h RO	**Reserved (RSVD_1):** Reserved
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 12 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG12.Region Base

27.175 Flash Region 13 (BIOS_FREG13)—Offset E4h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 13 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG13.Region Limit
15	0h RO	**Reserved (RSVD_1):** Reserved
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 13 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG13.Region Base

27.176 Flash Region 14 (BIOS_FREG14)—Offset E8h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 14 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG14.Region Limit
15	0h RO	**Reserved (RSVD_1):** Reserved
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 14 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG14.Region Base

27.177 Flash Region 15 (BIOS_FREG15)—Offset ECh

Access Method

Type:MEM Register
(Size: 32 bits)

Default:7FFFh

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Reserved (RSVD):** Reserved
30:16	0h RO/V	**Region Limit (RL):** This specifies address bits 26:12 for the Region 15 Limit. The value in this register is loaded from the contents in the Flash Descriptor.FLREG15.Region Limit
15	0h RO	**Reserved (RSVD_1):** Reserved
14:0	7FFFh RO/V	**Region Base (RB):** This specifies address bits 26:12 for the Region 15 Base. The value in this register is loaded from the contents in the Flash Descriptor.FLREG15.Region Base

27.178 RPMC SFDP Table (BIOS_RPMC0_D0)—Offset 108h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**RPMC SFDP Table dword 0 Device 0 (RPMC0_D0)**

27.179 RPMC SFDP Table (BIOS_RPMC1_D0)—Offset 10Ch

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**RPMC SFDP Table dword 1 Device 0 (RPMC1_D0)**

27.180 RPMC SFDP Table (BIOS_RPMC0_D1)—Offset 110h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**RPMC SFDP Table dword 0 Device 1 (RPMC0_D1)**

27.181 RPMC SFDP Table (BIOS_RPMC1_D1)—Offset 114h

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**RPMC SFDP Table dword 0 Device 1 (RPMC1_D1)**

27.182 BIOS Master Read Access Permissions (BIOS_BM_RAP)—Offset 118h

BM_RAP, BM_RAP registers are implemented in DNV, LBG and subsequent programs.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:C2h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:0	C2h RO/V	**BIOS Master Read Access Pemissions (BMRAP):** Each bit [15:0] corresponds to Regions [15:0]. If the bit is set, this master can read that particular region through register accesses. The contents of this register are that of the Flash Descriptor.Flash Master 1.Master Region Read Access OR a particular master has granted BIOS read permissions in their Master Read Access Grant register OR the Flash Descriptor Security Override strap is set. BIOS has read access to its own Region 1, 6 and 7 by default. Thus, the reset default of this field will be read as [quote]C2h[/quote].

27.183 BIOS Master Write Access Permissions (BIOS_BM_WAP)—Offset 11Ch

BM_RAP, BM_RAP registers are implemented in DNV, LBG and subsequent programs.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:42h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:0	42h RO/V	**BIOS Master Write Access Pemissions (BMWAP):** Each bit [15:0] corresponds to Regions [15:0]. If the bit is set, this master can erase and write that particular region through register accesses. The contents of this register are that of the Flash Descriptor.Flash Master 1.Master Region Write Access OR a particular master has granted BIOS write permissions in their Master Write Access Grant register OR the Flash Descriptor Security Override strap is set. BIOS has write access to its own Region 1 and 6 by default. Thus, the reset default of this field will be read as [quote]42h[/quote].

27.184 CSXE Flash Protected Range 0 (BIOS_CSXE_PR0)—Offset 184h

These are not physical registers in the Host memory space. These addresses provide the host with a read-only view of the values stored in the CSxE Protected Range registers.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.

27.185 CSXE Flash Protected Range 1 (BIOS_CSXE_PR1)—Offset 188h

These are not physical registers in the Host memory space. These addresses provide the host with a read-only view of the values stored in the CSxE Protected Range registers.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.

27.186 CSXE Flash Protected Range 2 (BIOS_CSXE_PR2)—Offset 18Ch

These are not physical registers in the Host memory space. These addresses provide the host with a read-only view of the values stored in the CSxE Protected Range registers.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.

27.187 CSXE Flash Protected Range 3 (BIOS_CSXE_PR3)—Offset 190h

These are not physical registers in the Host memory space. These addresses provide the host with a read-only view of the values stored in the CSxE Protected Range registers.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.

27.188 CSXE Flash Protected Range 4 (BIOS_CSXE_PR4)—Offset 194h

These are not physical registers in the Host memory space. These addresses provide the host with a read-only view of the values stored in the CSxE Protected Range registers.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the CSXE Flash Program Registers, PR0-PR4, WPR0.

27.189 Write Protected Range 0 (BIOS_CSXE_WPR0)—Offset 198h

This is not a physical register in the Host memory space. This address provides the Host with a read-only view of the values stored in the CSXE Write Protected Range register.

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** When set, this bit indicates that the Base and Limit fields in this register are valid and that writes and erases directed to addresses between them (inclusive) must be blocked by hardware. The base and limit fields are ignored when this bit is cleared.
30:16	0h RO/V	**Blocked Range Limit (BRL):** This field corresponds to FLA address bits 26:12 and specifies the upper limit of the protected range. Address bits 11:0 are assumed to be FFFh for the limit comparison. Any address greater than the value programmed in this field is unaffected by this protected range.
15	0h RO	**Reserved (RSVD):** Reserved.
14:0	0h RO/V	**Blocked Range Base (PRB):** This field corresponds to FLA address bits 26:12 and specifies the lower base of the protected range. Address bits 11:0 are assumed to be 000h for the base comparison. Any address less than the value programmed in this field is unaffected by this protected range.

27.190 IE Flash Protected Range 0 (BIOS_IE_PR0)—Offset 1B0h

These are not physical registers in the Host memory space. These addresses provide the host with a read-only view of the values stored in the IE Protected Range registers. <product specific = [quote]DNV, LBG[/quote]>

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.

27.191 IE Flash Protected Range 1 (BIOS_IE_PR1)—Offset 1B4h

These are not physical registers in the Host memory space. These addresses provide the host with a read-only view of the values stored in the IE Protected Range registers. <product specific = [quote]DNV, LBG[/quote]>

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.

27.192 IE Flash Protected Range 2 (BIOS_IE_PR2)—Offset 1B8h

These are not physical registers in the Host memory space. These addresses provide the host with a read-only view of the values stored in the IE Protected Range registers. <product specific = [quote]DNV, LBG[/quote]>

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.

27.193 IE Flash Protected Range 3 (BIOS_IE_PR3)—Offset 1BCh

These are not physical registers in the Host memory space. These addresses provide the host with a read-only view of the values stored in the IE Protected Range registers. <product specific = [quote]DNV, LBG[/quote]>

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.

27.194 IE Flash Protected Range 4 (BIOS_IE_PR4)—Offset 1C0h

These are not physical registers in the Host memory space. These addresses provide the host with a read-only view of the values stored in the IE Protected Range registers.
<product specific = [quote]DNV, LBG[/quote]>

Access Method

Type:MEM Register
(Size: 32 bits)

Default:0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Write Protection Enable (WPE):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.
30:16	0h RO/V	**Protected Range Limit (PRL):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.
15	0h RO/V	**Read Protection Enable (RPE):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.
14:0	0h RO/V	**Protected Range Base (PRB):** See description in the IE Flash Program Registers, PR0-PR4, WPR0.

27.195 eSPI Controller HW Status 1 (ESPI_HW_STS_2)—Offset 18h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD)**
23:16	0h RO/V	**EC Flash Rx Payload Size in Bytes (EC_FLASH_RX_PAYLOAD_SIZE):** The length specified in this field determines which bytes in the ESPI_EC_FLASH_RX_DATA_,0...15> registers contain valid received packet data. Set upon the completion of a EC Flash Rx packet transacation on eSPI bus; cleared when CSME FW clears the ESPI_MSI_STS.EC_FLASH_Avail bit.
15:12	0h RO	**Reserved (RSVD_1)**

Bit Range	Default & Access	Field Name (ID): Description
11:8	0h RO/V	**EC FLash Rx Tag (EC_FLASH_RX_TAG):** Tag value from the header in the eSPI Flash Rx packet. Tag may be used to track request-response handshakes by FW. It is not inspected or used by eSPI-MC.
7:6	0h RO	**Reserved (RSVD_2)**
5:4	0h RO/V	**EC Flash Rx Cmd (EC_FLASH_RX_CMD):** Command value form the header in the eSPI Flash Rx packet.
3:1	0h RO/V	**EC Flash Rx Status (EC_FLASH_RX_STATUS):** Set upon completion of a Flash Rx packet transaction on eSPI bus; cleared when CSME FW clears the ESPI_MSI_STS.EC_FLASH_REQ_Avail bit. This field is not set when an eSPI-MC HW serviced message is received from the Slave. Note that the ESPI_EC_FLASH_DATA[0..N] bytes are valid only if the EC_FLASH_Rx_Status is OK or Slave_Non_Fatal_Error. 3b000: Status not valid 3b001: Link Fatal Error Type 1 3b001: Reserved 3b010: Reserved 3b011: Malformed Slave Response Payload; Payload length > Max Payload Size or Read request size > Max Read Request Size [Fatal Error Type 2] 3b100: Slave Response Non_Fatal_Error 3b101 3b110: Reserved 3b111: OK
0	0h RO/V	**EC_FLASH Rx In Progress (EC_FLASH_RX_IN_PROGRESS):** Set when EC Flash Rx (Slave to eSPI/CSME) packet has grant on eSPI Bus bus, cleared when Rx packet is buffered in eSPI-MC and grant on eSPI bus is removed

27.196 (EC_FLASH_WR_DATA_0)—Offset 1A0h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**EC_FLASH_WR_DATA_0:** EC Flash write data recieved from the EC, with length (in bytes) specified in ESPI_HW_STS.OOB_SMBus_Rx_Payload_Size. When the data is valid, eSPI-MC asserts the Rx Avail MSI to CSME. Note: Data in these registers are invalid when the ESPI_HW_STS_0.OOB_Rx_in_Progress bit is set or when the ESPI_HW_STS_0.OOB_SMBus_Rx_Status field indicates an invalid/ error state

27.197 (EC_FLASH_WR_DATA_1)—Offset 1A4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**EC_FLASH_WR_DATA_1:** same as register EC FLASH WR DATA 0

27.198 (EC_FLASH_WR_DATA_2)—Offset 1A8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**EC_FLASH_WR_DATA_2:** same as register EC FLASH WR DATA 0

27.199 (EC_FLASH_WR_DATA_3)—Offset 1ACh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**EC_FLASH_WR_DATA_3:** same as register EC FLASH WR DATA 0

27.200 (EC_FLASH_WR_DATA_4)—Offset 1B0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**EC_FLASH_WR_DATA_4:** same as register EC FLASH WR DATA 0

27.201 (EC_FLASH_WR_DATA_5)—Offset 1B4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**EC_FLASH_WR_DATA_5:** same as register EC FLASH WR DATA 0

27.202 (EC_FLASH_WR_DATA_6)—Offset 1B8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**EC_FLASH_WR_DATA_6:** same as register EC FLASH WR DATA 0

27.203 (EC_FLASH_WR_DATA_7)—Offset 1BCh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**EC_FLASH_WR_DATA_7:** same as register EC FLASH WR DATA 0

27.204 (EC_FLASH_WR_DATA_8)—Offset 1C0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**EC_FLASH_WR_DATA_8:** same as register EC FLASH WR DATA 0

27.205 (EC_FLASH_WR_DATA_9)—Offset 1C4h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**EC_FLASH_WR_DATA_9:** same as register EC FLASH WR DATA 0

27.206 (EC_FLASH_WR_DATA_10)—Offset 1C8h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**EC_FLASH_WR_DATA_10:** same as register EC FLASH WR DATA 0

27.207 (EC_FLASH_WR_DATA_11)—Offset 1CCh

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**EC_FLASH_WR_DATA_11:** same as register EC FLASH WR DATA 0

27.208 (EC_FLASH_WR_DATA_12)—Offset 1D0h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**EC_FLASH_WR_DATA_12:** same as register EC FLASH WR DATA 0

27.209 (EC_FLASH_WR_DATA_13)—Offset 1D4h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**EC_FLASH_WR_DATA_13:** same as register EC FLASH WR DATA 0

27.210 (EC_FLASH_WR_DATA_14)—Offset 1D8h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**EC_FLASH_WR_DATA_14:** same as register EC FLASH WR DATA 0

27.211 (EC_FLASH_WR_DATA_15)—Offset 1DCh

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**EC_FLASH_WR_DATA_15:** same as register EC FLASH WR DATA 0

27.212 (EC_FLASH_REQ_ADDR)—Offset 1E0h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**EC_FLASH_REQ_ADDR:** same as register EC FLASH WR DATA 0

27.213 (EC_FLASH_CPL_CTL)—Offset 1E4h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S/V	**EC Flash CPL Done (EC_FLASH_CPL_DONE):** Set by CSME; cleared by eSPI-MC. CSME sets this bit after it has loaded the EC_FLASH_RD_DATA_xx registers and set the EC_FLASH_STS register.
30:0	0h RO	**Reserved (RSVD)**

27.214 (EC_FLASH_CPL_STS)—Offset 1E8h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD)**

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**EC Flash CPL Cycle Type (EC_FLASH_CPL_CYC_TYP):** CSME stes this field after it has completed the flash acces before setting the EC_FLASH_CPL_DONE bit.

27.215 (EC_FLASH_RD_DATA_0)—Offset 200h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**EC_FLASH_RD_DATA_0:** CSME to Slave. Flash read completion data loaded by CSME, with length (in bytes) specified in ESPI_HW_STS_2.EC_FLASH_REQ_Size.

27.216 (EC_FLASH_RD_DATA_1)—Offset 204h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**EC_FLASH_RD_DATA_1:** same as register EC FLASH RD DATA 0

27.217 (EC_FLASH_RD_DATA_2)—Offset 208h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**EC_FLASH_RD_DATA_2:** same as register EC FLASH RD DATA 0

27.218 (EC_FLASH_RD_DATA_3)—Offset 20Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**EC_FLASH_RD_DATA_3:** same as register EC FLASH RD DATA 0

27.219 (EC_FLASH_RD_DATA_4)—Offset 210h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**EC_FLASH_RD_DATA_4:** same as register EC FLASH RD DATA 0

27.220 (EC_FLASH_RD_DATA_5)—Offset 214h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**EC_FLASH_RD_DATA_5:** same as register EC FLASH RD DATA 0

27.221 (EC_FLASH_RD_DATA_6)—Offset 218h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**EC_FLASH_RD_DATA_6:** same as register EC FLASH RD DATA 0

27.222 (EC_FLASH_RD_DATA_7)—Offset 21Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**EC_FLASH_RD_DATA_7:** same as register EC FLASH RD DATA 0

27.223 (EC_FLASH_RD_DATA_8)—Offset 220h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**EC_FLASH_RD_DATA_8:** same as register EC FLASH RD DATA 0

27.224 (EC_FLASH_RD_DATA_9)—Offset 224h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**EC_FLASH_RD_DATA_9:** same as register EC FLASH RD DATA 0

27.225 (EC_FLASH_RD_DATA_10)—Offset 228h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**EC_FLASH_RD_DATA_10:** same as register EC FLASH RD DATA 0

27.226 (EC_FLASH_RD_DATA_11)—Offset 22Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**EC_FLASH_RD_DATA_11:** same as register EC FLASH RD DATA 0

27.227 (EC_FLASH_RD_DATA_12)—Offset 230h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**EC_FLASH_RD_DATA_12:** same as register EC FLASH RD DATA 0

27.228 (EC_FLASH_RD_DATA_13)—Offset 234h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**EC_FLASH_RD_DATA_13:** same as register EC FLASH RD DATA 0

27.229 (EC_FLASH_RD_DATA_14)—Offset 238h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**EC_FLASH_RD_DATA_14:** same as register EC FLASH RD DATA 0

27.230 (EC_FLASH_RD_DATA_15)—Offset 23Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**EC_FLASH_RD_DATA_15:** same as register EC FLASH RD DATA 0

27.231 Device and Vendor ID (CSXE_ESPI_DID_VID)—Offset 0h

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 31
Function: 0

Default: 31978086h

Bit Range	Default & Access	Field Name (ID): Description
31:16	3197h RO	**Device Identification (DID)**
15:0	8086h RO	**Vendor Identification (VID):** Indicates Intel

27.232 Device Control (CSXE_ESPI_STS_CMD)—Offset 4h

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 31
Function: 0

Default: 6100400h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**Detected Parity Error (DPE):** Not implemented. Hardwired to 0.
30	0h RO	**Signaled System Error (SSE):** Not implemented. Hardwired to 0.
29	0h RO	**Received Master Abort (RMA):** This bit must be set by a master device whenever its transaction (except for Special Cycle) is terminated with an Unsupported Request (UR) completion status (a.k.a. Master Abort). All master devices must implement this bit. eSPI does not initiate any upstream transactions to CSME system memory space.
28	0h RO	**Received Target Abort (RTA):** This bit must be set by a master device whenever its transaction is terminated with a Completer Abort (CA) completion status (a.k.a. Target Abort). All master devices must implement this bit. eSPI does not initiate any upstream transactions to CSME system memory space.
27	0h RO	**Signaled Target Abort (STA):** This bit must be set by a target device whenever it completes a Posted or Non-Posted transaction with a Completer Abort (CA) completion status (a.k.a.Target Abort). Devices that will never signal Target-Abort do not need to implement this bit. eSPI does not signal CA on any downstream Posted or Non-Posted transaction on the CSME root space.
26:25	3h RO	**Devsel Timing (DEVT)**
24	0h RO	**Master Data Parity Error (MDPE)**
23	0h RO	**Fast Back to Back Capable (FBTBC)**
22	0h RO	**Reserved (RSVD0)**
21	0h RO	**66 Mhz Capable (MCAP):** Not 66MHz Capable Device
20	1h RO	**Capablities List (CAPL):** This optional read-only bit indicates whether or not this device implements the pointer for a New Capabilities linked list at offset 34h. A value of zero indicates that no New Capabilities linked list is available. A value of one indicates that the value read at offset 34h is a pointer in Configuration Space to a linked list of new capabilities.
19	0h RO	**Interrupt Status (INTS):** This read-only bit reflects the state of the interrupt in the device/function. Only when the Interrupt Disable bit in the command register is a 0 and this Interrupt Status bit is a 1, will the device's/function's INTx# signal be asserted. Setting the Interrupt Disable bit to a 1 has no effect on the state of this bit.
18:11	0h RO	**Reserved (RSVD1)**

Bit Range	Default & Access	Field Name (ID): Description
10	1h RO	**Interrupt Disable (INTD):** This bit disables the device/function from asserting INTx#. A value of 0 enables the assertion of its INTx# signal. A value of 1 disables the assertion of its INTx# signal. This bit's state after RST# is 0. Read-only and hardwired to 1 for a device that does NOT support pin-based interrupt.
9	0h RO	**Fast Back to Back Enable (FBTBEN)**
8	0h RO	**System Error Enable (SERREN):** Not implemented. Hardwired to 0.
7	0h RO	**Reserved (RSVD)**
6	0h RO	**Parity Error Response (PERRR):** Not implemented. Hardwired to 0.
5	0h RO	**VGA Palette Snoop (VGAPS)**
4	0h RO	**Memory Write and Invalidate Enable (MWRIEN)**
3	0h RO	**Special Cycles (SPCYC)**
2	0h RW	**Bus Master Enable (BME):** Controls a device's ability to act as a master on the bus. A value of 0 disables the device from generating traffic. A value of 1 allows the device to behave as a bus master. State after RST# is 0. **Note:** This bit must be set to 1b to enable eSPI-MC to generate upstream MSI (posted write over IOSF-P) to CSME (in addition to ESPI_MSI_MC_NP_CID.MSIE).
1	0h RW	**Memory Space Enable (MSE):** Controls a device's response to Memory Space accesses. A value of 0 disables the device response. A value of 1 allows the device to respond to Memory Space accesses. State after RST# is 0. Typically, FW programs BAR while MSE is cleared and then enables MSE.
0	0h RO	**IO Space Enable (IOSE):** Controls a device's response to I/O Space accesses. A value of 0 disables the device response. A value of 1 allows the device to respond to I/O Space accesses. State after RST# is 0. Read-only and hardwired to 0 for a device that does NOT support I/O Space accesses.

27.233 Revision ID and Class Code (CSXE_ESPI_CC_RID)—Offset 8h

Access Method

Type: CFG Register (Size: 32 bits)
Device: 31
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Base Class Code (BCC):** Base Class Code
23:16	0h RO	**Sub-Class Code (SCC):** Sub-Class Code
15:8	0h RO	**Programming Interface (PI):** Programming Interface
7:0	0h RO/V	**Revision ID (RID):** Revision ID

27.234 CSXE BAR0 MMIO (CSXE_ESPI_MBAR)—Offset 10h

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 31
Function: 0

Default: 8h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**Memory BAR (MEMBAR):** Software programs this register with the base address of the device's memory region
11:4	0h RO	**Memory Size (MEMSIZE):** Hard wired to 0 to indicate 4KB of memory space
3	1h RO	**Prefetchable (PREFETCH):** Set to '1' to indicate there are no side-effects on reads
2:1	0h RO	**Type (TYP):** Set to '00' to allow mapping anywhere in 32-bit address space
0	0h RO	**Memory Space Indicator (MEMSPACE):** Set to 0 for Memory Space

27.235 Capabilities List Pointer (CSXE_ESPI_CAPP)—Offset 40h

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 31
Function: 0

Default: 44h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD)**

Bit Range	Default & Access	Field Name (ID): Description
7:0	44h RO	**Capabilities Pointer (CAPP):** Indicates the pointer for the first entry in the capabilities list

27.236 MSI Message Control, Next Pointer and Capability ID (CSXE_ESPI_MSIMC_MSINP_MSICID)—Offset 44h

Access Method

Type: CFG Register (Size: 32 bits)
Device: 31
Function: 0

Default: 5005h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD)**
24	0h RO	**Per Vector Masking Capable (PVMC):** This function does not support MSI per vector masking
23	0h RO	**64 bit address capable (XAC):** This function is not capable of sending 64 bit message address
22:20	0h RO	**Multiple Message Enable (MMEN):** Encoded number of interrupt vectors allocated by SW. Value of zero indicates one vector.
19:17	0h RO	**Multiple Message Capable (MMC):** Encoded number of interrupt vectors supported. Value of zero indicates one vector.
16	0h RW	**MSI Enable (MSIE):** If set (1) CXME MSI interrupt delivery is enabled. When this bit is cleared, prior to returning the configuration write completion, the device must send any pending MSI(s).
15:8	50h RO	**Next Item Pointer (NXTP):** Indicates the pointer for the next entry in the capabilities list. Hardwired to 50h to point to the PM Capability list.
7:0	5h RO	**Capability ID (CAPID):** Hardwired to 05h to indicate the linked list item as being the MSI Capability registers

27.237 MSI Message Address (CSXE_ESPI_MSIMA)—Offset 48h

MSI format is DW memory write in CSME root space with 32-bit addressing, 16 bit of data (upper word of data is zeros). The FW will only know the source (based on the FW programmed data field or vector) and it will need to read the cause/status register in device to get reason for MSI. FW clears after reading it.

Access Method

Type: CFG Register (Size: 32 bits)
Device: 31
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**Message Address (MADDR):** DW aligned MSI message address.
1:0	0h RO	**Reserved (RSVD)**

27.238 MSI Message Data (CSXE_ESPI_MSIMD)—Offset 4Ch

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 31
Function: 0

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD)**
15:0	0h RW	**Message Data (MDAT):** MSI Message Data

27.239 PCI Power Management Capability (CSXE_ESPI_PMCAP_PMNP_PMCID)—Offset 50h

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 31
Function: 0

Default: 40030001h

Bit Range	Default & Access	Field Name (ID): Description
31:27	8h RO	**PME Support (PMES):** This 5-bit field indicates the power states in which the function may assert PME#. A value of 0b for any bit indicates that the function is not capable of asserting the PME# signal while in that power state. bit(11) X XXX1b - PME# can be asserted from D0 bit(12) X XX1Xb - PME# can be asserted from D1 bit(13) X X1XXb - PME# can be asserted from D2 bit(14) X 1XXXb - PME# can be asserted from D3hot bit(15) 1 XXXXb - PME# can be asserted from D3cold
26	0h RO	**D2 Support (D2S):** This device does not support D2
25	0h RO	**D1 Support (D1S):** This device does not support D1

Bit Range	Default & Access	Field Name (ID): Description
24:22	0h RO	**Aux Current (AUXC):** Not Applicable
21	0h RO	**Device Specific Initialization (DSI):** Indicates whether special initialization of this function is required (beyond the standard PCI configuration header) before the generic class device driver is able to use it. A 1 indicates that the function requires a device specific initialization sequence following transition to the D0 uninitialized state. Hardwired to 0 to indicate NO Device Specific Initialization is required.
20	0h RO	**Reserved (RSVD)**
19	0h RO	**PME Clock (PMECLK):** Not Applicable
18:16	3h RO	**Version (VER):** Value of 011b indicates that this function complies with rev 1.2 of PCI Power Management Interface Spec
15:8	0h RO	**Next Item Pointer (NXTP):** Indicates the pointer for the next entry in the capabilities list. Hardwired to 0 to indicate no more linked list item.
7:0	1h RO	**Capability ID (CAPP):** Hardwired to 01h to indicate the linked list item as being the PCI Power Management registers

27.240 PCI Power Management Control and Status (CSXE_ESPI_PMD_PMCSRBSE_PMCSR)—Offset 54h

Access Method

Type: CFG Register
(Size: 32 bits)

Device: 31
Function: 0

Default: 8h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Data (DATA):** Not implemented. Hardwired to 0.
23:16	0h RO	**Reserved (RSVD)**

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C/V/P	**PME Status (PMESTS):** This bit is set when the function would normally assert the PME signal independent of the state of the PME_En bit. Writing a 1 to this bit will clear it and cause the function to stop asserting a PME (if enabled). Writing a 0 has no effect. If the function supports PME from D3cold, then this bit is sticky and must be explicitly cleared by the operating system each time the operating system is initially loaded
14:13	0h RO	**Data Scale (DS):** Not Applicable
12:9	0h RO	**Data Select (DSEL):** Not Applicable
8	0h RW	**PME Enable (PMEEN):** A 1 enables the function to assert PME. When 0, PME assertion is disabled. This bit defaults to 0 if the function does not support PME generation from D3cold. If the function supports PME from D3cold then this bit is sticky and must be explicitly cleared by the operating system each time it is initially loaded.
7:4	0h RO	**Reserved (RSVD1)**
3	1h RO	**No Soft Reset (NSR):** When set to 1, this bit indicates that devices transitioning from D3hot to D0 because of PowerState commands do not perform an internal reset. Configuration Context is preserved. Upon transition from the D3hot to the D0 Initialized state, no additional operating system intervention is required to preserve Configuration Context beyond writing the PowerState bits.
2	0h RO	**Reserved (RSVD2)**
1:0	0h RW	**Power State (PWRST):** This 2-bit field is used both to determine the current power state of a function and to set the function into a new power state. The definition of the field values is given below. 00b - D0 01b - D1 10b - D2 11b - D3hot If software attempts to write an unsupported, optional state to this field, the write operation must complete normally on the bus; however, the data is discarded and no state change occurs.

27.241 eSPI MSI cause (CSXE_ESPI_MSI_STS)—Offset 0h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RO	**Reserved (RSVD)**
5	0h RO/V	**EC Flash Load Done (EC_FLASH_LOAD_DONE):** This bit is set when a SLAVE_LOAD_DONE_VW is received from slave 0. This is an indication of: a. SPI controller has completed the strap pull, and b. eSPI-MC has enabled and configured the VW, OOB channel, and flash channel in MAF mode c. EC has completed loading its firmware
4	0h RW/1C/V	**EC Flash Request Avail (EC_FLASH_REQ_AVAIL):** This bit is set when a new EC Master Atttached Flash packet is received. ESPI_HW_STS_2 and ESPI_EC_FLASH_WR_DATA <0...1> registers contain the related flash packet data. The eSPI-MC will not accept any new OOB packets from the Slave until this bit is cleared when ESPI_MSI_EN.OOB_FW_STATE bit is 2'1X. The MSI generation is gated by the ESPI_MSI_EN.EC_FLASH_REQ_AVAIL_EN bit.
3	0h RW/1C/V	**OOB IOSF Unsupported Request (OOB_IOSF_URD):** Set to 1 by hardware upon detecting an Unsupported Request (UR) for a Posted Transaction on IOSF-P CSME Root Space that is not considered an Advisory Non-Fatal error. This MSI status is for notification only; it does not impact the Rx, Tx or processing of any HW serviced OOB packets. The MSI generation is gated by teh ESPI_MSI_EN.OOB_IOSF_URRE bit. Software must write a '1' in order to clear the state of this bit.
2	0h RW/1C/V	**OOB Hardware Serviced Packet Error (OOB_HWSVC_ERR):** This bit is set when an error is detected on the Rx/Tx of an OOB HW serviced packet. CSME FW can inspect and clear the appropriate ESPI_OOB_CTL_STS.OOB_HWSVC_*_ERROR_* register and take appropriate action, if necessary. This MSI is for notification only; it does not impact the Rx, Tx or processing of any HW serviced OOB packets. The MSI generation is gated by the ESPI_MSI_EN.OOB_HWSVC_Error_En bit.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW/1C/V	**OOB SMBus Sent (OOB_SMBUS_SENT):** This bit is set when a CSME-to-EC OOB packet has been transmitted to the EC on the eSPI bus. ESPI_HW_STS_0 contains related transmit status. eSPI-MC will not start transmit of the next CSME-to-EC OOB packet until this bit is cleared when ESPI_MSI_EN.OOB_FW_STATE bit is 2'1X. This bit is not set when a eSPI-MC HW serviced OOB messages to the slave. This bit does not gate any eSPI-MC HW serviced OOB messages to the slave. The MSI generation is gated by the ESPI_MSI_EN.OOB_SMBus_Sent_En bit.
0	0h RW/1C/V	**OOB SMBus Available (OOB_SMBUS_AVAIL):** This bit is set when a new EC-to-CSME OOB packet is received. ESPI_HW_STS_1 and ESPI_OOB_SMBUS_RX_DATA_<0..18> registers contain the related packet data. The eSPI-MC will not accept any new OOB packets (CSME or eSPI-MC HW serviced) from the slave until this bit is cleared when ESPI_MSI_EN.OOB_FW_STATE bit is 2'1X. This bit is not set when an eSPI-MC HW serviced OOB message is received from the slave. The MSI generation is gated by the ESPI_MSI_EN.OOB_SMBus_Avail_En bit.

27.242 eSPI MSI Enables (CSXE_ESPI_MSI_EN)—Offset 4h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RW	**OOB FW State (OOB_FW_STATE):** 00 - CSME FW is not ready. PME is disabled and all MSIs are masked. (CSME is in reset). 01 - CSME FW is supsended, PME is enabled and all MSIs are masked. (CSME is powergated). 10 - CSME FW is ready, PME is disabled, and all MSIs are masked. (CSME Active: Interrupt-Service / No-MSI Mode) 11 - CSME FW is ready, PME is disabled, and MSIs are unmasked. (CSME active).
29:5	0h RO	**Reserved (RSVD)**
4	0h RW	**EC Flash REQ Avail MSI Enable (EC_FLASH_REQ_AVAIL_MSI_EN):** MSI enable for the ESPI_MSI_STS.EC_FLASH_REQ_AVAIL status. Note: Depending on the setting of ESPI_OOB_CTL_STS.OOB_FW_STATE field, a MSI or a PME to the CSME may be generated when this bit is set

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**OOB IOSF URRE MSI Enable (OOB_IOSF_URRE_MSI_EN):** If set to 1 by software, it allows reporting of an Unsupported Request (UR) for a Posted transaction on IOSF-P CSME Root Space as a MSI. When such a transaction is detected, OOB_IOSF_URD bit is set and a MSI is sent to the CSME
2	0h RW	**OOB Hardware Serviced Packet Error MSI Enable (OOB_HWSVC_ERR_MSI_EN):** Enable for the OOB_HWSVC_ERR MSI.
1	0h RW	**OOB SM Bus Sent Enable Bit (OOB_SMBUS_SENT_EN):** MSI enable for the ESPI_MSI_STS.OOB_SMBus_Sent status. Refer to Note for bit [0] for additional details. Enable for the OOB_SMBus_Sent MSI. Note: eSPI-MC will not start transmit of the next OOB CSME-to-EC packet until this bit is set.
0	0h RW	**OOB SM Bus Available Enable Bit (OOB_SMBUS_AVAIL_EN):** MSI enable for the ESPI_MSI_STS.OOB_SMBus_Avail status. Note: Depending on the setting of ESPI_OOB_CTL_STS.OOB_FW_STATE field, a MSI or a PME to the CSME may be generated when this bit is set. Enable for the OOB_SMBus_Avail MSI. Note: eSPI-MC will not accept any new OOB packets until this bit is set when OOB_FW_Rdy is asserted

27.243 eSPI Controller HW Status 0 (CSXE_ESPI_HW_STS_0)—Offset 10h

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RO	**Reserved (RSVD)**
3:1	0h RO/V	**OOB SMBus Tx Status Bits (OOB_SMBUS_TX_STATUS):** Set upon completion of an OOB SMBus Tx packet transaction on eSPI bus,cleared when CSME FW clears the ESPI_MSI_STS.OOB_SMBus_Sent bit. 3'b000: Status not valid 3'b001: Fatal Error Type 1 3'b010 - 3'b011 : Reserved 3'b100: Slave Response Non_Fatal_Error 3'b101 3'b110: Reserved 3'b111: OK
0	0h RO/V	**OOB SMBus Tx In Progress (OOB_SMBUS_TX_IN_PROGRESS):** Set when OOB SMBus Tx CSME to Slave packet has grant on eSPI bus; cleared when Tx packet is sent and grant on eSPI bus is removed

27.244 eSPI Controller HW Status 1 (CSXE_ESPI_HW_STS_1)—Offset 14h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD)**
23:16	0h RO/V	**OOB SMBus Recieved payload size in Bytes (OOB_SMBUS_RX_PAYLOAD_SIZE):** Set upon completion of a OOB SMBus Rx packet transaction on eSPI bus, cleared when CSME FW clears the ESPI_MSI_STS.OOB_SMBus_Avail bit. This field is not set when an eSPI-MC HW serviced message is received from the Slave.
15:12	0h RO	**Reserved (RSVD_1)**
11:8	0h RO/V	**OOB SMBus Rx Tag (OOB_SMBUS_RX_TAG):** Tag value from the header in the eSPI OOB Rx packet. Tag can be potentially used to track request-response handshakes. It is not inspected or used by eSPI-MC.
7:6	0h RO	**Reserved (RSVD_2)**
5:4	0h RO/V	**OOB SMBus Recieve Slave ID (OOB_SMBUS_RX_SLVID):** Set upon completion of an OOB SMBus Rx packet transaction on eSPI bus; cleared when CSME FW clears the ESPI_MSI_STS.OOB_SMBus_Avail bit. This field is not set when an eSPI-MC HW serviced message is received from the Slave. 2'b00: OOB SMBus packet received from eSPI Slave 0 2'b01: OOB SMBus packet received from eSPI Slave 1 (Note: *Only* supported when a when a second eSPI Slave device is present) 2b10: Reserved 2b11: Reserved
3:1	0h RO/V	**OOB SMBus Recieve Status (OOB_SMBUS_RX_STATUS):** Set upon completion of an OOB SMBus Rx packet transaction on eSPI bus; cleared when CSME FW clears the ESPI_MSI_STS.OOB_SMBus_Avail bit. This field is not set when an eSPI-MC HW serviced message is received from the Slave. Note that the ESPI_RX_DATA[0..N] bytes are valid only if the OOB_SMBus_Rx_Status is OK or Slave_Non_Fatal_Error. 3b000: Status not valid 3b001: Fatal Error Type 1 3b010: Reserved 3b011: Rx_Invalid_Size (payload exceeds OOB_MPS) 3b100: Slave Response Non_Fatal_Error 3b101 3b110: Reserved 3b111: OK

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO/V	**OOB Rx In Progress (OOB_RX_IN_PROGRESS):** Set when OOB Rx (Slave to eSPI/CSME) packet has grant on eSPI Bus bus, cleared when Rx packet is buffered in eSPI-MC and grant on eSPI bus is removed

27.245 eSPI OOB Channel Controller and Status 0 (CSXE_ESPI_OOB_CTL_STS_0)—Offset 1Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD)**
15	0h RW/O	**Slave OOB Reset Ack Override (SLV1_OOB_RST_ACK_OVRD):** This bit has no impact since eSPI-MC does not support OOB_Reset_Ack VW for Slave 1.
14	0h RW/1C/V	**OOB SMBus Packet Dropped (SLV1_OOB_SMBUS_PKT_DROPPED):** 1 indicates that the eSPI-MC received one or more OOB messages from the Slave addressed to CSME FW prior to the CSME FW being ready or if the eSPI CSME function is in D3Hot set and the ESPI_PME_EN bit is cleared. This bit is sticky and CSMW FW can clear it by writing a 1 to it.
13	0h RO/V	**Slave Status OOB Avail (SLV1_STS_OOB_AVAIL):** Reflects the most recent state of the OOB_AVAIL bit in the Slave STATUS register. A 1 in this bit indicates that the Slave has a OOB message (either CSME FW or HW serviced) available to send to the master.
12	0h RO/V	**Slave Status OOB Free (SLV1_STS_OOB_FREE):** Reflects the most recent state of the OOB_FREE bit in the Slave STATUS register. A 1 in this bit indicates that the Slave has buffer space available to receive the next OOB packet from the eSPI Master (PCH).
11:9	0h RO/V	**OOB Max Packet Size (SLV1_OOB_MPS):** The negotiated Maximum Payload Size for the OOB channel supported by the eSPI-MC and Slave. Note that this size refers to the data payload itself and excludes the additional SMBus header/PEC bytes. The encodings are the same as in the eSPI Specification OOB Channel Slave Configuration Register.

Bit Range	Default & Access	Field Name (ID): Description
8	0h RO/V	**OOB Channel Supported (SLV1_OOB_CH_SUPPORTED):** Reflects whether the Slave supports the OOB Channel. If this is set to zero, it implies that the platform does not support any CSME EC communication in an eSPI enabled platform. Note: The SLV1_* bits in this register are *only* supported when a when a second eSPI Slave device is present
7	0h RW/O	**Slave OOB Reset Ack Override (SLV0_OOB_RST_ACK_OVRD):** A 1 in this bit will cause the eSPI-MC to not wait for the Slave OOB_RESET_ACK Virtual Wire before (immediately) asserting the ResetPrepAck (CSXE space, GenPrep). The OOB_Reset_Warn VW will be transmitted to the Slave independent of the setting for this bit. This is a Write Once bit and cleared only by a eSPI_RESET# assertion.
6	0h RW/1C/V	**OOB SMBus Packet Dropped (SLV0_OOB_SMBUS_PKT_DROPPED):** 1 indicates that the eSPI-MC received one or more OOB messages from the Slave addressed to CSME FW prior to the CSME FW being ready or if the eSPI CSME function is in D3Hot set and the ESPI_PME_EN bit is cleared. This bit is sticky and CSMW FW can clear it by writing a 1 to it.
5	0h RO/V	**Slave Status OOB Avail (SLV0_STS_OOB_AVAIL):** Reflects the most recent state of the OOB_AVAIL bit in the Slave STATUS register. A 1 in this bit indicates that the Slave has a OOB message (either CSME FW or HW serviced) available to send to the master.
4	0h RO/V	**Slave Status OOB Free (SLV0_STS_OOB_FREE):** Reflects the most recent state of the OOB_FREE bit in the Slave STATUS register. A 1 in this bit indicates that the Slave has buffer space available to receive the next OOB packet from the eSPI Master (PCH).
3:1	0h RO/V	**OOB Max Packet Size (SLV0_OOB_MPS):** The negotiated Maximum Payload Size for the OOB channel supported by the eSPI-MC and Slave. Note that this size refers to the data payload itself and excludes the additional SMBus header/PEC bytes. The encodings are the same as in the eSPI Specification OOB Channel Slave Configuration Register.
0	0h RO/V	**OOB Channel Supported (SLV0_OOB_CH_SUPPORTED):** Reflects whether the Slave supports the OOB Channel. If this is set to zero, it implies that the platform does not support any CSME EC communication in an eSPI enabled platform. Note: This bit reflects the RDY bit of the Slave's OOB Configuration register.

27.246 eSPI OOB Channel Controller and Status 1 (CSXE_ESPI_OOB_CTL_STS_1)—Offset 20h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**Reserved (RSVD)**
13:12	0h RO/V	**OOB Hardware Serviced Packet Tx Error Slave ID (OOB_HWSVC_TX_ERROR_SLVID):** Set upon the unsuccessful transmit of a non-CSME OOB message to the Slave. This field is valid only if the OOB_HWSVC_TX_ERROR_STATUS bit is set. 2'b00: Error on HWSVC packet transmitted to eSPI Slave 0 2'b01: Error on HWSVC packet transmitted to eSPI Slave 1 (Note: *Only* supported when a when a second eSPI Slave device is present) 2b10: Reserved 2b11: Reserved
11:9	0h RO/V	**OOB Hardware Serviced Packet Tx Error Cause (OOB_HWSVC_TX_ERROR_CAUSE):** Set upon the unsuccessful transmit of a non-CSME OOB message to the Slave. This field is valid only if the OOB_HWSVC_TX_ERROR_STATUS bit is set. Note: A Link Fatal Error Type 1 on any PUT_OOB command is reported in ESPI_HW_STS_1.OOB_SMBus_Tx_Status. 3'b000: Status not valid. 3'b001: Link Fatal Error Type 1 3'b010 - 3'b011: Reserved 3'b100: Slave Response Code: Non_Fatal_Error 3'b101 - 3'b111: Reserved
8	0h RO/V	**OOB Hardware Serviced Packet Tx Error Status (OOB_HWSVC_TX_ERROR_STATUS):** 1 indicates that the eSPI-MC detected errors on one or more HW serviced OOB messages to the Slave. Refer to Note for bit [0] for additional details.
7:6	0h RO	**Reserved (RSVD_2)**
5:4	0h RO/V	**OOB Hardware Serviced Packet Rx Error Slave ID (OOB_HWSVC_RX_ERROR_SLVID):** Set upon the unsuccessful receipt of a non-CSME OOB message from the slave. This field is valid only if the OOB_HWSVC_RX_ERROR_STATUS bit is set. 2'b00: Error on HWSVC packet received from eSPI Slave 0 2'b01: Error on HWSVC packet received from eSPI Slave 1 (Note: *Only* supported when a when a second eSPI Slave device is present) 2b10: Reserved 2b11: Reserved

Bit Range	Default & Access	Field Name (ID): Description
3:1	0h RO/V	**OOB Hardware Serviced Packet Rx Error Cause (OOB_HWSVC_RX_ERROR_CAUSE):** Set upon the unsuccessful receive of a non-CSME OOB message from the Slave. This field is valid only if the OOB_HWSVC_RX_ERROR_STATUS bit is set. 3'b000: Status not valid. 3'b001: Malformed Slave Response Payload: Invalid Slave Destination Address for a non-CSME message [Fatal Error Type 2]. 3'b010: Malformed Slave Response Payload: Invalid Command Code for a non-CSME message [Fatal Error Type 2]. 3'b011: Malformed Slave Response Payload: Payload length > Max Payload Size for HW serviced message Command Code [Fatal Error Type 2] 3'b100: Slave Response Code: Non_Fatal_Error 3'b101: Reserved 3'b110: Reserved 3'b111: Reserved
0	0h RO/V	**OOB Hardware Serviced Packet Rx Error Status (OOB_HWSVC_RX_ERROR_STATUS):** A 1 indicates that the eSPI-MC received one or more OOB messages with invalid payload from the Slave addressed to a non-CSME target. This bit is sticky and CSMW FW can clear it by writing a 1 to it. It does not cause a MSI to be generated to the CSME and does not affect the operation of the OOB channel. Note: When this bit is set, the ESPI_MSI_STS.OOB_HWSVC_Error bit is also set and a MSI/PME is generated depending on the state of the ESPI_MSI_EN.OOB_FW_STATE field. This bit is cleared when FW clears the ESPI_MSI_STS.OOB_HWSVC_Error bit. This state of this bit does not affect the servicing of subsequent OOB packets.

27.247 OOB SMBus Tx Control (CSXE_ESPI_OOB_SMBUS_TX_CTL)—Offset 2Ch

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1S/V	**OOB SMBus Tx Go (OOB_SMBUS_TX_GO):** Set by CSME; cleared by eSPI-MC upon OOB SMBus Tx packet transmission to the Slave being completed on the eSPI Bus, which can only happen after Slave had indicated OOB_FREE status. CSME sets this bit after it has loaded the ESPI_SMBUS_TX_DATA_xx registers and set the OOB_SMBus_Tx_Payload_Size in this register. eSPI-MC will set the ESPI_MSI_STS.OOB_SMBus_Sent bit (if ESPI_MSI_EN.OOB_SMBus_Sent_En bit is set) when the Slave has received the packet. Note: If ESPI_MSI_EN. OOB_SMBus_Sent_En bit is set then ESPI_MSI_STS.OOB_ SMBus_Sent bit must be cleared in order for this bit to take effect
30	0h RW/1S/V	**OOB SMBus Tx Abort (OOB_SMBUS_TX_ABORT):** Set by CSME; cleared by eSPI-MC upon invalidating a pending OOB SMBus Tx packet. This bit has no impact if the pending OOB SMBus Tx packet has already been arbitrated on to the eSPI bus. Writing a 1 to this bit also clears the OOB_SMBus_Tx_Go bit.
29:16	0h RO	**Reserved (RSVD_1)**
15:12	0h RW	**OOB SMBus Tx Tag (OOB_SMBUS_TX_TAG):** Tag value for the header in the eSPI OOB Tx packet. Tag can be potentially used to track request-response handshakes. It is not inspected or used by eSPI-MC.
11:10	0h RO	**Reserved (RSVD_0)**
9:8	0h RW	**OOB SMBus Transmit Slave ID (OOB_SMBUS_TX_SLVID):** This field specifies the eSPI Slave device that is the target of the OOB SMBus Tx packet. 2'b00: eSPI Slave 0 is the target of OOB SMBus Tx packet 2'b01: eSPI Slave 1 is the target of OOB SMBus Tx packet (Note: *Only* supported when a when a second eSPI Slave device is present) 2b10: Reserved 2b11: Reserved
7:0	0h RW	**OOB SMBus Tx Payload Size (OOB_SMBUS_TX_PAYLOAD_SIZE):** Payload size in bites

27.248 (CSXE_ESPI_OOB_SMBUS_TX_DATA_0)—Offset 30h

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**OOB_SMBUS_TX_DATA_0:** CSME to Slave SMBusTx packet loaded by CSME, with length (in bytes) specified in ESPI_OOB_SMBus_TX_CTL.OOB_SMBus_Tx_Payload_Size. eSPIMC transmits these bytes to the EC upon a subsequent write to ESPI_OOB_TX_CTL register. Note: These registers should not be written to as long as the ESPI_OOB_TX_CTL.OOB_Tx_Go bit is asserted (it is set by CSME and cleared by eSPI-MC,otherwise it will result in undefined behavior

27.249 (CSXE_ESPI_OOB_SMBUS_TX_DATA_1)—Offset 34h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**OOB_SMBUS_TX_DATA_1:** same as register ESPI OOB SMBUS TX DATA 0

27.250 (CSXE_ESPI_OOB_SMBUS_TX_DATA_2)—Offset 38h

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**OOB_SMBUS_TX_DATA_2:** same as register ESPI OOB SMBUS TX DATA 0

27.251 (CSXE_ESPI_OOB_SMBUS_TX_DATA_3)—Offset 3Ch

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**OOB_SMBUS_TX_DATA_3:** same as register ESPI OOB SMBUS TX DATA 0

27.252 (CSXE_ESPI_OOB_SMBUS_TX_DATA_4)—Offset 40h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**OOB_SMBUS_TX_DATA_4:** same as register ESPI OOB SMBUS TX DATA 0

27.253 (CSXE_ESPI_OOB_SMBUS_TX_DATA_5)—Offset 44h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**OOB_SMBUS_TX_DATA_5:** same as register ESPI OOB SMBUS TX DATA 0

27.254 (CSXE_ESPI_OOB_SMBUS_TX_DATA_6)—Offset 48h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**OOB_SMBUS_TX_DATA_6:** same as register ESPI OOB SMBUS TX DATA 0

27.255 (CSXE_ESPI_OOB_SMBUS_TX_DATA_7)—Offset 4Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**OOB_SMBUS_TX_DATA_7:** same as register ESPI OOB SMBUS TX DATA 0

27.256 (CSXE_ESPI_OOB_SMBUS_TX_DATA_8)—Offset 50h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**OOB_SMBUS_TX_DATA_8:** same as register ESPI OOB SMBUS TX DATA 0

27.257 (CSXE_ESPI_OOB_SMBUS_TX_DATA_9)—Offset 54h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**OOB_SMBUS_TX_DATA_9:** same as register ESPI OOB SMBUS TX DATA 0

27.258 (CSXE_ESPI_OOB_SMBUS_TX_DATA_10)—Offset 58h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**OOB_SMBUS_TX_DATA_10:** same as register ESPI OOB SMBUS TX DATA 0

27.259 (CSXE_ESPI_OOB_SMBUS_TX_DATA_11)—Offset 5Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**OOB_SMBUS_TX_DATA_11:** same as register ESPI OOB SMBUS TX DATA 0

27.260 (CSXE_ESPI_OOB_SMBUS_TX_DATA_12)—Offset 60h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**OOB_SMBUS_TX_DATA_12:** same as register ESPI OOB SMBUS TX DATA 0

27.261 (CSXE_ESPI_OOB_SMBUS_TX_DATA_13)—Offset 64h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**OOB_SMBUS_TX_DATA_13:** same as register ESPI OOB SMBUS TX DATA 0

27.262 (CSXE_ESPI_OOB_SMBUS_TX_DATA_14)—Offset 68h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**OOB_SMBUS_TX_DATA_14:** same as register ESPI OOB SMBUS TX DATA 0

27.263 (CSXE_ESPI_OOB_SMBUS_TX_DATA_15)—Offset 6Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**OOB_SMBUS_TX_DATA_15:** same as register ESPI OOB SMBUS TX DATA 0

27.264 (CSXE_ESPI_OOB_SMBUS_TX_DATA_16)—Offset 70h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**OOB_SMBUS_TX_DATA_16:** same as register ESPI OOB SMBUS TX DATA 0

27.265 (CSXE_ESPI_OOB_SMBUS_TX_DATA_17)—Offset 74h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**OOB_SMBUS_TX_DATA_17:** same as register ESPI OOB SMBUS TX DATA 0

27.266 (CSXE_ESPI_OOB_SMBUS_TX_DATA_18)—Offset 78h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**OOB_SMBUS_TX_DATA_18:** same as register ESPI OOB SMBUS TX DATA 0

27.267 (CSXE_ESPI_OOB_SMBUS_RX_DATA_0)—Offset 150h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**OOB_SMBUS_RX_DATA_0:** Slave to CSME SMBus Rx packet received from the EC, with length (in bytes)specified in ESPI_HW_STS. OOB_SMBus_Rx_Payload_Size. When the data is valid, eSPI-MC asserts the Rx_Avail MSI to CSME. Note: Data in these registers is invalid when the ESPI_HW_STS_0.OOB_Rx_in_Progress bit is set or when the ESPI_HW_STS_0.OOB_SMBus_Rx_Status field indicates an invalid,error state

27.268 (CSXE_ESPI_OOB_SMBUS_RX_DATA_1)—Offset 154h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**OOB_SMBUS_RX_DATA_1**

27.269 (CSXE_ESPI_OOB_SMBUS_RX_DATA_2)—Offset 158h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**OOB_SMBUS_RX_DATA_2**

27.270 (CSXE_ESPI_OOB_SMBUS_RX_DATA_3)—Offset 15Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**OOB_SMBUS_RX_DATA_3**

27.271 (CSXE_ESPI_OOB_SMBUS_RX_DATA_4)—Offset 160h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**OOB_SMBUS_RX_DATA_4**

27.272 (CSXE_ESPI_OOB_SMBUS_RX_DATA_5)—Offset 164h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**OOB_SMBUS_RX_DATA_5**

27.273 (CSXE_ESPI_OOB_SMBUS_RX_DATA_6)—Offset 168h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**OOB_SMBUS_RX_DATA_6**

27.274 (CSXE_ESPI_OOB_SMBUS_RX_DATA_7)—Offset 16Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**OOB_SMBUS_RX_DATA_7**

27.275 (CSXE_ESPI_OOB_SMBUS_RX_DATA_8)—Offset 170h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**OOB_SMBUS_RX_DATA_8**

27.276 (CSXE_ESPI_OOB_SMBUS_RX_DATA_9)—Offset 174h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**OOB_SMBUS_RX_DATA_9**

27.277 (CSXE_ESPI_OOB_SMBUS_RX_DATA_10)—Offset 178h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**OOB_SMBUS_RX_DATA_10**

27.278 (CSXE_ESPI_OOB_SMBUS_RX_DATA_11)—Offset 17Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**OOB_SMBUS_RX_DATA_11**

27.279 (CSXE_ESPI_OOB_SMBUS_RX_DATA_12)—Offset 180h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**OOB_SMBUS_RX_DATA_12**

27.280 (CSXE_ESPI_OOB_SMBUS_RX_DATA_13)—Offset 184h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**OOB_SMBUS_RX_DATA_13**

27.281 (CSXE_ESPI_OOB_SMBUS_RX_DATA_14)—Offset 188h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**OOB_SMBUS_RX_DATA_14**

27.282 (CSXE_ESPI_OOB_SMBUS_RX_DATA_15)—Offset 18Ch

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**OOB_SMBUS_RX_DATA_15**

27.283 (CSXE_ESPI_OOB_SMBUS_RX_DATA_16)—Offset 190h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**OOB_SMBUS_RX_DATA_16**

27.284 (CSXE_ESPI_OOB_SMBUS_RX_DATA_17)—Offset 194h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**OOB_SMBUS_RX_DATA_17**

27.285 (CSXE_ESPI_OOB_SMBUS_RX_DATA_18)—Offset 198h

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**OOB_SMBUS_RX_DATA_18**

§ §

28 CNVi Registers

This chapter documents the registers in Bus: 0, Device 12, Function 0.

28.1 CNVI_WIFI_VEN_DEV_ID — Offset 0h

identifies the manufacturer of the device(vendor) and type.(Offset 000 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 0h	8086 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**DEV_ID:** Device ID identifies the particular PCI device
15:0	8086h RO	**VEN_ID:** vendor ID

28.2 CNVI_WIFI_PCI_COM_STAT — Offset 4h

command and status register .(Offset 004 h) HARDWIRED --]] HARDWIRED

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 4h	100000 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C	**DET_PAR_ERR:** Detected Parity Error
30	0h RW/1C	**SIG_SYS_ERR:** Signaled System Error
29	0h RW/1C	**REC_MAS_ABRT:** Received Master Abort
28	0h RW/1C	**REC_TAR_ABRT:** Received Target Abort
27	0h RW/1C	**SIG_TAR_ABRT:** Signaled Target Abort
26:25	0h RO	**DEVSEL_TIMING:** DEVSEL Timing, does not apply to PCI Express HARDWIRED
24	0h RW/1C	**MAS_DATA_PAR_ER:** Master Data Parity Error
23	0h RO	**FAST_BTB_TCAP:** Fast Back-to-Back Transaction Capable, does not apply to PCI Express HARDWIRED

Bit Range	Default & Access	Field Name (ID): Description
22	0h RO	**Reserved (RSVD):** Reserved
21	0h RO	**OLF_FREQ_CAP:** 66 MHz Capable, does not apply to PCI Express HARDWIRED
20	1h RO	**CAP_LST:** Capability List, must be set to 1
19	0h RO	**INTRPT_STS:** Interrupt status, reflects the state of the interrupt in the device
18:11	0h RO	**Reserved (RSVD):** Reserved
10	0h RW	**INTRPT_DIS:** Interrupt Disable, controls the ability of the device to generate legacy interrupt messages
9	0h RO	**FAST_BTB_TNSCEN:** Fast Back-to-Back Transaction Enable, does not apply to PCI Express HARDWIRED
8	0h RW	**SERR_EN:** SERR Enable, when set 1, the device can drive the SERR# line
7	0h RO	**IDSEL_STEP_W_CY:** IDSEL Stepping/Wait Cycle Control, does not apply to PCI Express HARDWIRED
6	0h RW	**PAR_ERR:** Parity Error Enable
5	0h RO	**VGA_PALT_SNOOP:** VGA Palette Snoop, does not apply to PCI Express HARDWIRED
4	0h RO	**MEM_WR_INVALD:** Memory Write and Invalidate, does not apply to PCI Express HARDWIRED
3	0h RO	**SPEC_CYC_ENB:** Special Cycle Enable, does not apply to PCI Express HARDWIRED
2	0h RW	**BUS_MAS:** Bus Master Enable
1	0h RW	**MEM_SP_ACC:** Memory Space access enable
0	0h RO	**IO_SPC_AC_EN_0:** IO Space access enable

28.3 CNVI_WIFI_PCI_CLASS_CODE — Offset 8h

revision identifier and class code(Offset 008 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 8h	2800000 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	2h RO	**SE_CLASS:** Base Class, classifies the type of function the device per form

Bit Range	Default & Access	Field Name (ID): Description
23:16	80h RO	**SUB_CLASS:** Sub-Class, identifies more specifically the function of the device
15:8	0h RO	**INTERFACE:** Interface, identifies a specific register-level programming interface
7:0	0h RO	**REV_ID:** Revision ID identifies the revision of particular PCI device

28.4 CNVI_WIFI_PCI_BUS_COM — Offset Ch

revision identifier and class code(Offset 00C h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + Ch	800000 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**BIST_CAPB:** BIST Capable, BIST is implemented but not controlled by Configuration HARDWIRED
30	0h RO	**START_BIST:** Start BIST HARDWIRED
29:28	0h RO	**Reserved (RSVD):** Reserved
27:24	0h RO	**BIST_COMPL_CODE:** BIST Completion Code HARDWIRED
23	1h RO	**MUL_FUNC_DEV:** Multifunction Device. is used to identify multifunction device
22:16	0h RO	**PRFIND_HEAD_LAY:** Predefined Header Layout HARDWIRED
15:8	0h RO	**LATNCY_TIMR:** Latency Timer, Does not apply to PCI Express HARDWIRED
7:0	0h RW	**CACH_LN_SIZE:** Cache Line Size

28.5 CNVI_WIFI_BAR0 — Offset 10h

Base Address Register BAR0 Low (Offset 010 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 10h	4 h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RW	**BS_ADD_FLD_RW:** Base Address Field
13:4	0h RO	**BS_ADD_FLD_RO:** Base Address Field HARDWIRED
3	0h RO	**PREFETCHBL:** Prefetchable, defines the block of memory as Prefetchable or not
2:1	2h RO	**DECOD_WDTH_FLD:** Decoder Width Field (10b-64bit reg) HARDWIRED
0	0h RO	**MEM_SPAC_INDIC:** Memory Space Indicator HARDWIRED

28.6 CNVI_WIFI_BAR1 — Offset 14h

Base Address Register BAR0 High (Offset 014 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 14h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**BS_ADD_FLD:** Base Address Field

28.7 CNVI_WIFI_BAR2 — Offset 18h

Reserved hard coded to 0 (Offset 018 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 18h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**DAT_RESERVED:** Reserved hard coded to 0

28.8 CNVI_WIFI_BAR3 — Offset 1Ch

Reserved hard coded to 0 (Offset 01C h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 1Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**DAT_RESERVED:** Reserved hard coded to 0

28.9 CNVI_WIFI_BAR4 — Offset 20h

Reserved hard coded to 0 (Offset 020 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 20h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**DAT_RESERVED:** Reserved hard coded to 0

28.10 CNVI_WIFI_BAR5 — Offset 24h

Reserved hard coded to 0 (Offset 024 h) cooment: common to all functions

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 24h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**DAT_RESERVED:** Reserved hard coded to 0

28.11 CNVI_WIFI_CIS_PTR — Offset 28h

Reserved hard coded to 0 (Offset 028 h) Optional, is implemented by devices that share silicon between CardBus and PCI, points to the Card Information Structure (CIS) the CardBus card, contains the offset of the CIS

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 28h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**CARDB_PTR:** CardBus CIS Pointer

28.12 CNVI_WIFI_SUBSYS_ID — Offset 2Ch

Identifies the manufacturer of the device(vendor) and type.(Offset 02C h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 2Ch	8086 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**SB_DEV_ID:** Subsystem (device) ID
15:0	8086h RO	**SB_VEN_ID:** Subsystem Vendor ID

28.13 CNVI_WIFI_ROM_BAR — Offset 30h

Expansion ROM Base Address (Offset 030 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 30h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**EXPN_ROM_ADD:** Expansion ROM Base Address HARDWIRED
10:1	0h RO	**Reserved (RSVD):** Reserved
0	0h RO	**EXPN_ROM_ENB:** Expansion ROM Enable HARDWIRED

28.14 CNVI_WIFI_CAP_PTR — Offset 34h

capabilities pointer register (Offset 034 h) bit [7:0] Point to the first item in the list of capabilities, provides an offset in the device's PCI Configuration Space for the location of the first item in the Capabilities List

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 34h	C8 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD):** Reserved
7:0	C8h RO	**CARDB_PTR:** Capabilities Pointer

28.15 CNVI_WIFI_RES_ADDRESS — Offset 38h

Reserved hard coded to 0 (Offset 038 h) Optional, is implemented by devices that share silicon between CardBus and PCI, points to the Card Information Structure (CIS) the CardBus card, contains the offset of the CIS

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 38h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**DAT_RESERVED:** Reserved

28.16 CNVI_WIFI_INTERRUPT — Offset 3Ch

revision identifier and class code(Offset 03C h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 3Ch	100 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**MAX_LATNC:** Max_Lat , Does not apply to PCI Express HARDWIRED
23:16	0h RO	**MIN_GNT:** Min_Gnt , Does not apply to PCI Express HARDWIRED
15:8	1h RO	**INT_PIN:** Interrupt Pin , Tells which interrupt pin the device uses For IOSF - option to write once
7:0	0h RW	**INT_LINE:** Interrupt Line, id which input interrupt request pin is routed to

28.17 CNVI_WIFI_GIO_CAP — Offset 40h

PCI Express Capabilities Register. (Offset 040 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 40h	928010 h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD):** Reserved
29:25	0h RO	**INTRPT_MSG_NUM:** Interrupt Message Number HARDWIRED
24	0h RO	**SLT_IMPLNT:** Slot Implemented HARDWIRED
23:20	9h RO	**DEV_POR_TYP:** Device/Port Type
19:16	2h RO	**CAP_VER:** Capability Version (0x2 at gen 2.0)
15:8	80h RO	**GIO_CAP_NXT_OFS:** The offset to the next PCI capability structure HARDWIRED
7:0	10h RO	**INCD_PCIE_CST:** indicates PCI Express Capability Structure HARDWIRED

28.18 CNVI_WIFI_GIO_DEV_CAP — Offset 44h

The Device Capabilities register identifies PCI Express device specific capabilities.(Offset 044 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 44h	10000EC0 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**Reserved (RSVD):** Reserved
28	1h RO	**FUNC_LVL_RES:** Function Level Reset Capability [too add GEN_2]
27:26	0h RO	**SLT_PW_LSCL:** Captured Slot Power Limit Scale
25:18	0h RO	**SLT_PW_LVAL:** Captured Slot Power Limit Value
17:16	0h RO	**Reserved (RSVD):** Reserved
15	0h RO	**ROLE_BASED_ERR:** This field indicates that the device support Error reporting
14:12	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
11:9	7h RO	**L1_ACC_LAT:** Endpoint L1 Acceptable Latency
8:6	3h RO	**L0S_ACC_LAT:** Endpoint L0s Acceptable Latency
5	0h RO	**EX_TAG_FIELD:** Extended Tag Field Supported
4:3	0h RO	**PHAN_FUNCS:** Phantom Functions Supported
2:0	0h RO	**MAX_PL_SIZE:** Max_Payload_Size Supported

28.19 CNVI_WIFI_GIO_DEV — Offset 48h

Device Control Register. (Offset 048 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 48h	100C10 h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RO	**Reserved (RSVD):** Reserved
21	0h RO	**TRANS_PEND:** indicates that a device has Non-Posted Requests which have not been completed
20	1h RO	**AUX_P_DET:** device that requires AUX power reports this bit (asynch signal)
19	0h RW/1C	**UNSOP_REQ_DET:** indicates that the device received Unsupported Request
18	0h RW/1C	**FAT_ERR_DET:** indicates status of fatal errors detected
17	0h RW/1C	**NFAT_ER_DET:** indicates status of non-fatal errors detected
16	0h RW/1C	**COR_ERR_DET:** indicates status of correctable errors detected
15	0h RO	**INIT_FNC_LV_RS:** Initiate Function Level Reset - A write of 1b initiates Function (init FLR)
14:12	0h RW	**MAX_RDRQ_SIZE:** sets the maximum Read Request size
11	1h RW	**EN_NO_SNOOP:** Enable No Snoop (if set device is permitted to set No Snoop bit)
10	1h RW	**AUX_PM_EN:** Auxiliary (AUX) Power PM Enable. sticky value.
9	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
8	0h RO	**IO_SPC_AC_EN_8:** IO Space access enable
7:5	0h RW	**MAX_PAY_SIZE:** sets maximum TLP payload size for the device functions
4	1h RW	**EN_REL_ORD:** when set device permitted to set the Relaxed Ordering bit
3	0h RW	**UNSOP_REQ_REP:** enables reporting of Unsupported Request when set
2	0h RW	**FAT_ERR_REP:** controls reporting of fatal errors
1	0h RW	**NFAT_ER_REP:** controls reporting of non-fatal errors
0	0h RW	**COR_ERR_REP:** controls reporting of correctable errors

28.20 CNVI_WIFI_GIO_LNK_CAP — Offset 4Ch

The Link Capabilities register identifies PCI Express Link specific capabilities.(Offset 04C h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 4Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**DAT_RESERVED:** Reserved

28.21 CNVI_WIFI_GIO_LINK — Offset 50h

Link Control and Link status Register.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 50h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**DAT_RESERVED:** RESERVED for IOSF

28.22 CNVI_WIFI_GIO_DEV_CAP_2 — Offset 64h

device control 2 (and status) register. (Offset 064 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 64h	80812 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RO	**Reserved (RSVD):** Reserved
19:18	2h RO	**OBFF_MEC_SUP:** OBFF suported : 0x00 -unsupported , 0x01 -sup using message signaling A, 0x10 -sup using signaling B ,0x11-supported using WAKE signaling
17:12	0h RO	**Reserved (RSVD):** Reserved
11	1h RO	**LTR_MEC_SUP:** LTR Mechanism Supported - a value of 1b indicates support for LTR.
10:5	0h RO	**Reserved (RSVD):** Reserved
4	1h RO	**CMP_TO_DIS_SUP:** 1- support for the Completion Timeout Disable, 0- not supported hardwired to 0x1
3:0	2h RO	**CMP_TO_RNG_SUP:** Completion Timeout Ranges Supported hardwired to 0x2

28.23 CNVI_WIFI_GIO_DEV_2 — Offset 68h

Device control (and status) register. (Offset 068 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 68h	5 h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD):** Reserved
14:13	0h RW	**OBFF_MEC_ENA:** OBFF enable: 0x00 -disable , 0x01 - enabled using message signaling A, 0x10 - message signaling B ,0x11-enabled using WAKE signaling
12:11	0h RO	**Reserved (RSVD):** Reserved
10	0h RW	**LTR_MEC_EN:** LTR Mechanism Enable - When Set enables the LTR.
9:5	0h RO	**Reserved (RSVD):** Reserved
4	0h RW	**CMP_TO_DIS:** When Set, this bit disables the Completion Timeout mechanism
3:0	5h RW	**CMP_TO_VAL:** this field allows system SW to modify the Completion Timeout value

28.24 CNVI_WIFI_GIO_LINK_2 — Offset 70h

link control 2 (and status) register (Offset 070 h) at this reg bits of type RWS - allow software to write sticky values to register.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 70h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**DAT_RESERVED:** RESERVED for IOSF

28.25 CNVI_WIFI_MSIX_CAP_HEAD — Offset 80h

MSIX Capability .(Offset 080 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 80h	F0011 h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**MSIX_ENABLE:** If set to 1, the function is permitted to use MSI-X to request service. System configuration software sets this bit to enable MSI-X. A device driver is prohibited from writing this bit to mask a function's service request. If 0, the function is prohibited from using MSI-X to request service.
30	0h RW	**FUN_MASK:** If set to 1, all of the vectors associated with the function are masked, regardless of their per-vector Mask bit states. If 0, each vector's Mask bit determines whether the vector is masked or not. Setting or clearing the MSI-X Function Mask bit has no effect on the state of the per-vector Mask bits
29:27	0h RO	**Reserved (RSVD):** Reserved
26:16	Fh RO	**TABLE_SIZE:** System software reads this field to determine the MSI-X Table Size N, which is encoded as (N-1). Wifi Host supports Table Size of 16 and encodes a number of 15 (Pulsar).
15:8	0h RO	**NEXT_PTR:** Pointer to the next item in the capabilities list. NULL if last
7:0	11h RO	**MSIX_CAP_ID:** The value of 11h in this field identifies the function as being MSI-X capable.

28.26 CNVI_WIFI_MSIX_TABLE_OFFSET — Offset 84h

MSIX Capability Structure - Table Offset Reg.(Offset 084 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 84h	2000 h

Bit Range	Default & Access	Field Name (ID): Description
31:3	400h RO	**TABLE_OFFSET:** Used as an offset from the address contained by the Base Address register to point to the base of the MSI-X Table. The lower 3 bits are masked off (set to zero) by software to form a 32-bit QWORD-aligned offset. In Wifi Host the table is at 0x2000, so the value of this field in 0x400.
2:0	0h RO	**TABLE_BIR:** Indicates which one of a function's Base Address registers, located beginning at 10h in configuration Space, and is used to map the function's MSI-X Table into Memory Space. Wifi Host cluster uses a single BAR to point to the MSI-X structures.

28.27 CNVI_WIFI_MSIX_PBA_OFFSET — Offset 88h

MSIX Capability Structure - Pending Bit Array Offset Reg.(Offset 088 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 88h	3000 h

Bit Range	Default & Access	Field Name (ID): Description
31:3	600h RO	**TABLE_OFFSET:** Used as an offset from the address contained by the Base Address register to point to the base of the MSI-X PBA Table. The lower 3 bits are masked off (set to zero) by software to form a 32-bit QWORD-aligned offset. In Wifi Host the table is at 0x3000, so the value of this field in 0x600.
2:0	0h RO	**TABLE_BIR:** Indicates which one of a function's Base Address registers, located beginning at 10h in configuration Space, and is used to map the function's MSI-X Table into Memory Space. Wifi Host cluster uses a single BAR to point to the MSI-X structures.

28.28 CNVI_WIFI_PMC — Offset C8h

Power Management Capabilities Register(Offset 0C8 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + C8h	C823D001 h

Bit Range	Default & Access	Field Name (ID): Description
31:27	19h RO	**PME_SUPRT:** PME Support,indicates the power states in which the device may assert PME
26	0h RO	**D2_PWR_MANG:** D2 Power Management State support
25	0h RO	**D1_PWR_MANG:** D1 Power Management State support
24:22	0h RO	**AUX_CUR:** AUX Current (Used data register instead)
21	1h RO	**DEV_SPC_INT:** Device Specific Initialization

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**Reserved (RSVD):** Reserved
19	0h RO	**PME_CLK:** PME Clock, does not apply to PCI Express HARDWIRED
18:16	3h RO	**VERSION:** value indicates that this function complies with the Revision 1.2
15:8	D0h RO	**PMC_NXT_PTR:** Next PTR, pointing to the location of next item in the functions capability list HARDWIRED
7:0	1h RO	**PMC_CAP_ID:** Capability ID, Indicates the linked list item is the PCI Power Management Registers HARDWIRED

28.29 CNVI_WIFI_PMCSR — Offset CCh

Power Management Status and Control (Offset 0CC h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + CCh	D000008 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	Dh RO	**PWR_DIS_CON:** used to report power consumption and heat dissipation (default for D3-0x1)
23	0h RO	**BUS_PWR_CLK_CEN:** Bus Power/Clock Control Enable, does not apply to PCI Express HARDWIRED
22	0h RO	**B2_B3_SUPRT:** B2/B3 Support, does not apply to PCI Express HARDWIRED
21:16	0h RO	**Reserved (RSVD):** Reserved
15	0h RW/1C	**PME_STAT:** This bit reflects whether the function has experienced a PME. sticky value.
14:13	0h RO	**DAT_SCALE:** Data Scale
12:9	0h RW	**DAT_SEL:** Data Select, selects the data value to be viewed through the Data register
8	0h RW	**PME_ENA:** PME Enable. sticky value.
7:4	0h RO	**Reserved (RSVD):** Reserved
3	1h RO	**NO_SOFT_RESET:** No_Soft_Reset
2	0h RO	**Reserved (RSVD):** Reserved
1:0	0h RW	**PWR_STATE:** Power State

28.30 CNVI_WIFI_MSI_MSG_CTRL — Offset D0h

Capability ID and Message Control.(Offset 0D0 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + D0h	804005 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23	1h RO	**BITA_CAP_64:** 64 bit address capable HARDWIRED
22:20	0h RO	**MUL_MSG_ENB:** Multiple Message Enable HARDWIRED
19:17	0h RO	**MUL_MSG_CAP:** Multiple Message Capable HARDWIRED
16	0h RW	**MSI_ENA:** function is enabled to use MSI to request service and is forbidden to use its interrupt pin
15:8	40h RO	**MSI_MC_NXT_PTR:** Next Capability pointer offset HARDWIRED
7:0	5h RO	**MSI_MC_CAP_ID:** Capability ID HARDWIRED

28.31 CNVI_WIFI_MSI_LOW_ADD — Offset D4h

Specifies the lower DWORD of the address for the MSI memory write transaction (Offset 0D4 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + D4h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:2	0h RW	**MSG_ADD_LOW:** lower DWORD of the address
1:0	0h RO	**MSI_LOW_AD_1_0:** HARDWIRED

28.32 CNVI_WIFI_MSI_HIGH_ADD — Offset D8h

Specifies the upper DWORD of the address for the MSI memory write transaction (Offset 0D8 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + D8h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**MSG_ADD_UP:** upper DWORD of the address

28.33 CNVI_WIFI_MSI_DATA — Offset DCh

Written by the system to indicate the lower 16 bits of the data written in the MSI memory write DWORD transaction (Offset 0DC h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + DCh	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD):** Reserved
15:0	0h RW	**MSG_DATA:** data written in the MSI memory write DWORD transaction

28.34 CNVI_WIFI_GIO_SERIAL_CAP — Offset 100h

Device Serial Number Capability.(Offset 100 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 100h	14C00000 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	14Ch RO	**NXT_CAP_OFF:** Next Capability Offset, next is the LTR header
19:16	0h RO	**CAP_VER:** Capability Version: Zero to skip Serial Number
15:0	0h RO	**EXT_CAP_ID:** PCI Express Extended Capability ID: Zero to skip Serial Number

28.35 CNVI_WIFI_GIO_SERIAL_LOW — Offset 104h

Serial number low.(Offset 104 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 104h	FF000000 h

Bit Range	Default & Access	Field Name (ID): Description
31:24	FFh RO	**HARDWIRED_SR_LW:** hardwired to 0xFF
23:16	0h RO	**BYTE_6:** byte 6 from OTP
15:8	0h RO	**BYTE_5:** byte 5 from OTP
7:0	0h RO	**BYTE_4:** byte 4 from OTP

28.36 CNVI_WIFI_GIO_SERIAL_UP — Offset 108h

Serial number up.(Offset 108 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 108h	FF h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**BYTE_1:** byte 1 from OTP
23:16	0h RO	**BYTE_2:** byte 2 from OTP
15:8	0h RO	**BYTE_3:** byte 3 from OTP
7:0	FFh RO	**HARDWIRED_SR_UP:** hardwired to 0xFF

28.37 CNVI_WIFI_UNCORRECT_ERR_SEV — Offset 10Ch

Uncorrectable Error severity Register. (Offset 10C h) at this reg bits of type RWS - the error is reported as fatal when the field is set to 1, g if it is cleared to 0, the error is considered non-fatal

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 10Ch	462031 h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (RSVD):** Reserved
25	0h RW	**TLP_PRFIX_BLOCKED_SVR:** TLP Prefix Blocked Error Severity (Optional) - report is not implemented. sticky value
24	0h RW	**ATMC_EGRS_BLOCKED_SVR:** AtomicOps Egress Blocked Severity (Optional) - report is not implemented. sticky value
23	0h RW	**MC_BLOCKED_TLP_SVR:** MC Blocked TLP Severity (Optional) - report is not implemented. sticky value
22	1h RW	**UNCORCT_INT_ERR_SVR:** Uncorrectable Internal Error Severity (Optional) - report is not implemented. sticky value
21	0h RW	**ACS_VIOL_SVR:** ACS Violation Severity (Optional) - report is not implemented. sticky value
20	0h RW	**UNSPR_REQ_ERR_SVR:** Unsupported Request Error severity sticky value
19	0h RO	**ECRC_ERR_SVR_19:** ECRC Error severity
18	1h RW	**MAL_TLP_SVR:** Malformed TLP severity . sticky value
17	1h RW	**REC_OVRF_SVR:** Receiver Overflow severity . sticky value
16	0h RW	**UNXPL_COM_SVR:** Unexpected Completion severity . sticky value
15	0h RW	**COM_AB_SVR:** Completer Abort severity . sticky value
14	0h RW	**COM_TO_SVR:** Completion Timeout severity. sticky value
13	1h RW	**FLWCNT_PR_SVR:** Flow Control Protocol Error severity. sticky value
12	0h RW	**POIS_TLP_SVR:** Poisoned TLP severity . sticky value
11:6	0h RO	**Reserved (RSVD):** Reserved
5	1h RW	**SURPRISE_DWN_SVR:** Surprise Down Error Severity (Optional) . sticky value
4	1h RW	**DLNK_PRERR_SVR:** Data Link Protocol Error severity . sticky value
3:1	0h RO	**Reserved (RSVD):** Reserved
0	1h RO	**TRNG_ERR_SVR_0:** Training Error severity

28.38 CNVI_WIFI_CORRECT_ERR_STAT — Offset 110h

PCI Express Capabilities Register. (Offset 110 h) at this reg bits of type RW1CS - , if is set to 1 indicates that the error occurred; g software may clear the bit by writing a 1

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 110h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**Reserved (RSVD):** Reserved
13	0h RW	**ADV_NON_FATAL:** Advisory Non Fatal Error
12	0h RW/1C	**REPL_TO_ST:** Replay Timer Timeout Error status . sticky value
11:9	0h RO	**Reserved (RSVD):** Reserved
8	0h RW/1C	**REPL_ROLVR_ST:** REPLAY_NUM Rollover status . sticky value
7	0h RW/1C	**BD_DLLP_ST:** Bad DLLP status . sticky value
6	0h RW/1C	**BD_TLP_ST:** Bad TLP status . sticky value
5:1	0h RO	**Reserved (RSVD):** Reserved
0	0h RW/1C	**REV_ERR_ST:** Receiver Error status . sticky value

28.39 CNVI_WIFI_CORRECT_ERR_MASK — Offset 114h

PCI Express Capabilities Register. (Offset 114 h) at this reg bits of type RWS - if set to 1 the error is masked.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 114h	2000 h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**Reserved (RSVD):** Reserved
13	1h RW	**ADV_NON_FATAL:** Advisory Non Fatal Error -- masked by default
12	0h RW	**REPL_TO_MSK:** Replay Timer Timeout Error mask . sticky value
11:9	0h RO	**Reserved (RSVD):** Reserved
8	0h RW	**REPL_ROLVR_MSK:** REPLAY_NUM Rollover mask . sticky value

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**BD_DLLP_MSK:** Bad DLLP mask . sticky value
6	0h RW	**BD_TLP_MSK:** Bad TLP mask . sticky value
5:1	0h RO	**Reserved (RSVD):** Reserved
0	0h RW	**REV_ERR_MSK:** Receiver Error mask . sticky value

28.40 CNVI_WIFI_ADVANCED_ERR_CAP — Offset 118h

Advanced Error Capabilities and Control register. (Offset 118 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 118h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RO	**Reserved (RSVD):** Reserved
8	0h RO	**ECRC_CHK_EN:** ECRC Check Enable, if set enables ECRC checking
7	0h RO	**ECRC_CHK_CP:** ECRC Check Capable, indicates the device is capable of checking ECRC
6	0h RO	**ECRC_GEN_EN:** ECRC Generation Enable, if set enables ECRC generation
5	0h RO	**ECRC_GEN_CP:** ECRC Generation Capable, indicates that the device is capable of generation ECRC
4:0	0h RO	**FRST_ERR_PNT:** error reported in the Uncorrectable Error Status register First Error Pointer, identifies bit position of the first

28.41 CNVI_WIFI_HEADER_LOG1 — Offset 11Ch

captures the header of TLP associated with error. (Offset 11C h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 11Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**HEADER_LOG_1:** captures the header of TLP associated with error

28.42 CNVI_WIFI_HEADER_LOG2 — Offset 120h

captures the header of TLP associated with error. (Offset 120 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 120h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**HEADER_LOG_2:** captures the header of TLP associated with error

28.43 CNVI_WIFI_HEADER_LOG3 — Offset 124h

captures the header of TLP associated with error. (Offset 124 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 124h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**HEADER_LOG_3:** captures the header of TLP associated with error

28.44 CNVI_WIFI_HEADER_LOG4 — Offset 128h

captures the header of TLP associated with error. (Offset 128 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 128h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**HEADER_LOG_4:** captures the header of TLP associated with error

28.45 CNVI_WIFI_LTR_EXTND_CAP_HEAD — Offset 14Ch

header of LTR extended capability reg.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 14Ch	16410018 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	164h RO	**LTR_HD_NXT_PTR:** Next Capability Offset. next is the Vendor Specific Capability Header
19:16	1h RO	**LTR_HD_CAP_VER:** Capability Version. must be 1 for this version
15:0	18h RO	**LTR_HD_CAP_ID:** PCI Express Extended Capability ID for the LTR Extended Capability is 0018h

28.46 CNVI_WIFI_LTR_MAX_SNOOP_NOSNOOP_LAT — Offset 150h

this register specifies the maximum nosnoop latency that a device is permitted to request.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 150h	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**Reserved (RSVD):** Reserved
28:26	0h RW	**NSNP_MX_LAT_SCL:** no-snoop max latency scale. set by SW
25:16	0h RW	**NSNP_MX_LAT_VAL:** no-snoop max latency value. set by SW
15:13	0h RO	**Reserved (RSVD):** Reserved
12:10	0h RW	**SNP_MAX_LAT_SCL:** snoop max latency scale. set by SW
9:0	0h RW	**SNP_MAX_LAT_VAL:** snoop max latency value. set by SW

28.47 CNVI_WIFI_L1PM_SUB_EXTND_CAP_HEAD — Offset 154h

L1_SUBSTATES extended capability header.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 154h	1001E h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RO	**NXT_PTR:** Next Capability Offset. 0x0 for last extended cap
19:16	1h RO	**CAP_VER:** Capability Version. must be 1 for this version
15:0	1Eh RO	**EXT_CAP_ID:** PCI Express Extended Capability ID for L1 PM

28.48 CNVI_WIFI_L1PM_SUB_CAP — Offset 158h

L1_SUBSTATES capability register.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 158h	481E1F h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD):** Reserved
23:19	9h RO	**L1SUB_PWRON_REQ:** Port requirement for T_POWER_ON value
18	0h RO	**Reserved (RSVD):** Reserved
17:16	0h RO	**T_PWRON_SCALE:** Portrequirement for T_POWER_ON scale
15:8	1Eh RO	**CMN_RESTORE_TM:** Port common mode restore time
7:5	0h RO	**Reserved (RSVD):** Reserved
4	1h RO	**L1SUB_PM_SUP:** L1 PM substate supported from OTP
3	1h RO	**L11_ASPM_SUP:** ASPM L1.1 is supported from OTP
2	1h RO	**L12_ASPM_SUP:** ASPM L1.2 is supported from OTP
1	1h RO	**L11_PM_SUP:** PM L1.1 is supported from OTP
0	1h RO	**L12_PM_SUP:** PM L1.2 is supported from OTP

28.49 CNVI_WIFI_L1PM_SUB_CNTRL — Offset 15Ch

L1_SUBSTATES control register.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 15Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RW	**LTR_L1_THRS_SCL:** LTR L1.2 threshold scale
28:26	0h RO	**Reserved (RSVD):** Reserved
25:16	0h RW	**LTR_L1_THRS_VAL:** LTR L1.1 threshold value, detemines if entry to L1 results in L1.1 or L1.2 (if they are enabled). scale of value set by: LTR L1.2 threshold scale
15:4	0h RO	**Reserved (RSVD):** Reserved
3	0h RW	**L11_ASPM_EN:** L1.1 ASPM is enabled
2	0h RW	**L12_ASPM_EN:** L1.2 ASPM is enabled
1	0h RW	**L11_PM_EN:** L1.1 PM is enabled
0	0h RW	**L12_PM_EN:** L1.2 PM is enabled

28.50 CNVI_WIFI_L1PM_SUB_CNTRL2 — Offset 160h

L1_SUBSTATES control register.

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 160h	28 h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD):** Reserved
7:3	5h RW	**T_POWER_ON_VAL:** T power on value. Along with power on scale - sets min time in L1.2.exit
2	0h RO	**Reserved (RSVD):** Reserved
1:0	0h RW	**L12_PWRON_SCALE:** T power on scale. 0 =2us, 1=10us, 2=100us, 11 = reserved

28.51 CNVI_WIFI_VEN_SPEC_CAP — Offset 164h

Vendor Specific Capability Header.(Offset 164 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 164h	1000B h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RO	**VSNEXT:** Next Capability Offset (VSNEXT): This field contains the offset to the next PCI Express capability structure or 000h if no other items exist in the linked list of capabilities. For Extended Capabilities implemented in device Configuration Space, this offset is relative to the beginning of PCI-compatible Configuration Space and thus must always be either 000h (for terminating list of capabilities) or greater than 0FFh. Next Capability Offset, next is the Device Serial Number Cap
19:16	1h RO	**VSECREV:** Vendor Specific Extended Capability Revision (VSECREV): the version of the PCIe capability structure present. Must be 1h for this version of the specification.
15:0	Bh RO	**VSPCIECID:** Vendor Specific PCI Express Extended Capability ID (VSPCIECID): The Extended Capability ID for the Vendor-Specific Capability is 000Bh.

28.52 CNVI_WIFI_VEN_SPEC_EXTND_CAP — Offset 168h

Vendor Specific Extended Capability Register.(Offset 168 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 168h	1400010 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	14h RO	**VSECLEN:** Vendor Specific Extended Capability Length (VSECLEN): This field indicates the # of bytes of this Vendor Specific capability inclusive of the 1st DW which includes the capability ID and Next pointer. For DevIdle, this is 14h
19:16	0h RO	**VSECREV:** Vendor Specific Extended Capability Revision (VSECREV): For this revision of DevIdle, this field is 0h
15:0	10h RO	**VSECID:** Vendor Specific Extended Capability ID (VSECID): DevIdle has been assigned the Intel VSEC ID of 10h

28.53 CNVI_WIFI_LTP_PTR — Offset 16Ch

SW LTR pointer.(Offset 16C h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 16Ch	0 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RO	**SWLTRLOC:** SW LTR Update MMIO Offset Location This register contains the location pointer to the SW LTR register in MMIO space, as an offset from the specified BAR. The value of this register is a don't care, if the Valid bit is not set

Bit Range	Default & Access	Field Name (ID): Description
19:4	0h RO	**Reserved (RSVD):** Reserved
3:1	0h RO	**BARNUM:** Base Address Register Number. Contains the 0's based AR Number of the BAR which contains the location of the SW LTR MMIO register.
0	0h RO	**VALID:** Set to '1' to indicate that the function has implemented a SW LTR register as specified in the DevIdle that can be located using the SWLTRLOC register and BARNUM. Set to '0' to indicate that the function has not implemented a SW LTR register that is compliant to the DevIdle definition. This could be because the function has not implemented SW LTR at all, or has a device specific version of SW LTR. Need to be fixed to 1'b0

28.54 CNVI_WIFI_DEV_IDLE_PTR — Offset 170h

DevIdle Pointer.(Offset 170 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 170h	31800001 h

Bit Range	Default & Access	Field Name (ID): Description
31:20	318h RO	**DEVIDLELOC:** DevIdle MMIO Offset Location.Contains the location pointer to the DevIdle register in MMIO space, as an offset from the specified BAR. The value of this register is a don't care, if the Valid bit is not set
19:4	0h RO	**Reserved (RSVD):** Reserved
3:1	0h RO	**BARNUM:** Base Address Register Number. Contains the 0's based BAR Number of the BAR which contains the location of the DevIdle MMIO register. When counting BAR Numbers, all BARs, regardless of size and type, consume one BAR BAR Number. Bar 0 is the 1st BAR. The value of this register is a don't care, if the Valid bit is not set. Fixed to 3'b0
0	1h RO	**VALID:** Set to '1' to indicate that the function has implemented a DevIdle register as specified in the DevIdle that can be located using the DEVIDLELOC register and BARNUM. Set to '0' to indicate that the function has not implemented a DevIdle register that is compliant to the DevIdle definition. This could be because the function has not implemented DevIdle at all, or has a device specific version of DevIdle. set to fix 1'b1

28.55 CNVI_WIFI_DEV_IDLE_PWR — Offset 174h

DevIdle Power on latency.(Offset 174 h)

Type	Size	Offset	Default
CFG	32 bit	[B:0, D:12, F:0] + 174h	800 h

Bit Range	Default & Access	Field Name (ID): Description
31:13	0h RO	**Reserved (RSVD):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
12:10	2h RO	**POLS:** Power On Latency Scale. Latency Scale multiplier: 010: 1us, 011: 32us All other settings are reserved. The value of this register is multiplied with the Power On Latency Value (POLV) to provide the DevIdle power on exit latency multiplier scale from 1us to 32ms.
9:0	0h RO	**POLV:** Power On Latency Value. 10-bit value that is multiplied by the Power On Latency Scale. A value of 0 indicates a power on latency of less than 1us.

§ §

29 GPIO Registers

29.1 Revision ID (REV_ID)—Offset 0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 940000h

Bit Range	Default & Access	Field Name (ID): Description
31:16	94h RO	**Hardware Base Revision ID (HWBASEREVID):** This field is used to indicate the baseline hardware revision. 0091h = RTL release for HAS0.91
15:0	0h RO	**Hardware Sub Revision ID (HWSUBIREVID):** This field is used to indicate the revision of the hardware derived from the baseline.

29.2 Capability List Register (CAP_LIST_0)—Offset 4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD_0):** Reserved
23:16	0h RO	**Capability Identification (CAPID):** A unique identification for the current capability. A value of 0 is reserved, and must not be used by any capability. Note:This field is always 0 for the first Capability List register.
15:0	0h RO	**Next Capability List Pointer (NXTCAPLPTR):** Specify the DW-aligned Pointer/ Address to the next item in this capabilities list and must be 0 if there is no capability at all or this is the last capability in the list.

29.3 Family Base Address (FAMBAR)—Offset 8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 300h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15:0	300h RO	**Family Base Address (FAMBAR):** This field provides the starting byte-align address of Family0 register sets within a Community. It is meant for software to discover from where the very first Family register (i.e. Family0 register) starts to compute the next Families address offsets.

29.4 Pad Base Address (PADBAR)—Offset Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 600h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15:0	600h RO	**Pad Base Address (PADBAR):** This field provides the starting byte-align address of Pad0 register sets within a Community. It is meant for software to discover from where the very first Pad register (i.e. Pad0 register) starts to compute the next Pad address offsets.

29.5 Miscellaneous Configuration (MISCCFG)—Offset 10h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: E043200h

Bit Range	Default & Access	Field Name (ID): Description
31:24	Eh RW	**GPIO Driver Mode Interrupt Select (GPDMINTSEL):** IRQ globally for all pads (GPI_IS with corresponding GPI_IE enable). 0 = Interrupt Line 0 1 = Interrupt Line 1 ... 255 = Interrupt Line 255
23:20	0h RW	**Reserved (Reserved1):** Reserved
19:16	4h RW	**GPIO Group to GPE_DW2 assignment encoding (GPE0_DW2):** This register assigns a specific GPIO Group to the ACPI GPE0[95:64]. The setting is SoC specific.
15:12	3h RW	**GPIO Group to GPE_DW1 assignment encoding (GPE0_DW1):** This register assigns a specific GPIO Group to the ACPI GPE0[63:32]. The setting is SoC specific.
11:8	2h RW	**GPIO Group to GPE_DW0 assignment encoding (GPE0_DW0):** This register assigns a specific GPIO Group to the ACPI GPE0[31:0]. The setting is SoC specific.
7:6	0h RW	**Reserved (Reserved2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**GPIO IOSF Sideband Dynamic Partition Clock Gating Enable (GPSIDEDPCGEN):** Specify whether the GPIO Community IOSF sideband clock should take part in partition clock gating 0 = Disable participation in IOSF sideband clock dynamic partition clock gating 1 = Enable participation in IOSF sideband clock dynamic partition clock gating
4	0h RW	**GPIO RCOMP clock Dynamic Local Clock Gating (GPRCOMPCDLCGEN):** Specify whether the GPIO Community should perform local clock gating on RCOMP clock 0 = Disable dynamic RCOMP clock local clock gating 1 = Enable dynamic RCOMP clock local clock gating
3	0h RW	**GPIO RTC clock Dynamic Local Clock Gating (GPRTCDLCGEN):** Specify whether the GPIO Community should perform local clock gating on RTC clock 0 = Disable dynamic RTC clock local clock gating 1 = Enable dynamic RTC clock local clock gating
2	0h RW	**GSX Static Local Clock Gating (GSXSLCGEN):** Specify whether the GSX controller should be statically clock gated for power saving if it is not enabled (even though the capability is available). 0 = Disable static local clock gating 1 = Enable static local clock gating
1	0h RW	**GPIO Dynamic Partition Clock Gating Enable (GPDPCGEN):** Specify whether the GPIO Community should take part in partition clock gating 0 = Disable participation in dynamic partition clock gating 1 = Enable participation in dynamic partition clock gating
0	0h RW	**GPIO Dynamic Local Clock Gating Enable (GPDLCGEN):** Specify whether the GPIO Community should perform local clock gating 0 = Disable dynamic local clock gating 1 = Enable dynamic local clock gating

29.6 Miscellaneous Secured Configuration (MISCSECCFG)—Offset 14h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD_0):** Reserved
7:4	0h RW	**Reserved (Reserved1):** Reserved
3	0h RW/1C/V	**Soft strap pull status bit (SSTRAPPULLSTAT):** Indicate the status of the last soft strap pull operation. If the implementation guarantee the success of the operation, then this bit can be hard-wired to 0. 0 = No error 1 = Error FW writes 1 to this bit to clear. This bit is set to 1 when soft strap pull request has been completed with completion without data (for both successful and unsuccessful completion status) or completion with data but with unsuccessful completion status. This bit will also set to 1 if there is any failure on the security and error checking in the fuse puller such as invalid SAI and etc. Fuse puller will assert puller_error in this case and this will set this bit to 1.
2	0h RW/1C/V	**Soft strap re-pull done bit (SSTRAPPULLDONE):** Indication if the soft strap pull has finished the operation 0 = operation is not done. Either not started or in progress 1 = operation is done/finish. FW writes 1 to this bit to clear.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Soft strap re-pull request bit (SSTRAPREPULLREQ):** Specify if fuse puller need to re-pull the soft strap. The SoC/PMC/FW can set this bit to request the community fuse puller to re-pull the soft strap. 0 to 1 transition = soft strap re-pull request trigger. A soft strap re-pull request will not trigger IP_READY message sent as the IP_READY message shall be sent during the initial soft strap pull by HW as part of the reset sequences. The SSTRAPREPULLREQ, SSTRAPPULLDONE and SSTRAPPULLSTAT are only meaningful for community with soft strap pull capability.
0	0h RW	**Event Trigger IOSF-SB Message Initiation Disable (SBTRIGDIS):** This bit disables the following list of IOSF-SB Message initiation. It does not disable the event trigger. When this bit is cleared to '0', any pending message due to event trigger while this bit is '1' shall be sent on IOSF-SB. Event list: 'Virtual Wire' message IRQ GPE/SCI NMI SMI

29.7 Reserved (RSVD0[0])—Offset 18h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.8 Reserved (RSVD0[1])—Offset 1Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.9 Pad Ownership (PAD_OWN_audio_0)—Offset 20h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_163):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_162):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

Bit Range	Default & Access	Field Name (ID): Description
23:22	0h RO	**Reserved (RSVD_2):** Reserved
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_161):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_160):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

Bit Range	Default & Access	Field Name (ID): Description
15:14	0h RO	**Reserved (RSVD_4):** Reserved
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_159):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_158):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

Bit Range	Default & Access	Field Name (ID): Description
7:6	0h RO	**Reserved (RSVD_6):** Reserved
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_157):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_156):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.10 Pad Ownership (PAD_OWN_audio_1)—Offset 24h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_171):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_170):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_169):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_168):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_167):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_166):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_165):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_164):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.11 Pad Ownership (PAD_OWN_audio_2)—Offset 28h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
29:28	0h RW	**Pad Ownership (PAD_OWN_vGPIO_34):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved
25:24	0h RW	**Pad Ownership (PAD_OWN_vGPIO_33):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
21:20	0h RW	**Pad Ownership (PAD_OWN_vGPIO_32):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved
17:16	0h RW	**Pad Ownership (PAD_OWN_vGPIO_31):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_175):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_174):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_173):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_172):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.12 Pad Ownership (PAD_OWN_audio_3)—Offset 2Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RO	**Reserved (RSVD_0):** Reserved
13:12	0h RW	**Pad Ownership (PAD_OWN_vGPIO_38):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW	**Pad Ownership (PAD_OWN_vGPIO_37):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_2):** Reserved
5:4	0h RW	**Pad Ownership (PAD_OWN_vGPIO_36):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Pad Ownership (PAD_OWN_vGPIO_35):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.13 Reserved (RSVD1[0])—Offset 30h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.14 Reserved (RSVD1[1])—Offset 34h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.15 Reserved (RSVD1[2])—Offset 38h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.16 Reserved (RSVD1[3])—Offset 3Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.17 Reserved (RSVD1[4])—Offset 40h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.18 Reserved (RSVD1[5])—Offset 44h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.19 Reserved (RSVD1[6])—Offset 48h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.20 Reserved (RSVD1[7])—Offset 4Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.21 Reserved (RSVD1[8])—Offset 50h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.22 Reserved (RSVD1[9])—Offset 54h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.23 Reserved (RSVD1[10])—Offset 58h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.24 Reserved (RSVD1[11])—Offset 5Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.25 GPI Virtual Wire Message Enable (GPI_VWE_audio_0)—Offset 60h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD_0):** Reserved
27	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_38):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
26	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_37):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
25	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_36):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
24	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_35):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
23	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_34):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
22	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_33):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
21	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_32):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_31):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
19	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_175):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
18	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_174):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
17	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_173):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
16	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_172):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
15	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_171):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
14	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_170):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_169):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
12	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_168):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_167):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_166):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
9	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_165):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
8	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_164):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_163):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_162):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
5	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_161):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
4	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_160):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_159):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_158):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_157):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_156):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.26 Reserved (RSVD2[0])—Offset 64h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.27 Reserved (RSVD2[1])—Offset 68h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.28 Reserved (RSVD2[2])—Offset 6Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.29 Reserved (RSVD2[3])—Offset 70h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.30 Reserved (RSVD2[4])—Offset 74h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.31 Reserved (RSVD2[5])—Offset 78h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.32 Reserved (RSVD2[6])—Offset 7Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.33 Pad Configuration Lock (PADCFGLOCK_audio_0)—Offset 80h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD_0):** Reserved
27	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_38):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_37):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
25	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_36):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
24	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_35):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
23	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_34):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
22	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_33):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
21	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_32):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_31):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
19	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_175):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_174):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
17	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_173):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_172):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
15	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_171):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_170):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
13	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_169):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_168):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
11	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_167):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_166):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
9	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_165):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_164):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
7	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_163):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_162):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
5	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_161):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_160):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
3	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_159):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_158):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
1	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_157):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_156):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.34 Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_audio_0)—Offset 84h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_38):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
26	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_37):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_36):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
24	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_35):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_34):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
22	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_33):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_32):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
20	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_31):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_175):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
18	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_174):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_173):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
16	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_172):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_171):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
14	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_170):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_169):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
12	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_168):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_167):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
10	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_166):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_165):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
8	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_164):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_163):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
6	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_162):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_161):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
4	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_160):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_159):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
2	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_158):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_157):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
0	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_156):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.35 Reserved (RSVD3[0])—Offset 88h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.36 Reserved (RSVD3[1])—Offset 8Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.37 Reserved (RSVD3[2])—Offset 90h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.38 Reserved (RSVD3[3])—Offset 94h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.39 Reserved (RSVD3[4])—Offset 98h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.40 Reserved (RSVD3[5])—Offset 9Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.41 Reserved (RSVD3[6])—Offset A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.42 Reserved (RSVD3[7])—Offset A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.43 Reserved (RSVD3[8])—Offset A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.44 Reserved (RSVD3[9])—Offset ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.45 Host Software Pad Ownership (HOSTSW_OWN_audio_0)—Offset B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD_0):** Reserved
27	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_38):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_37):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
25	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_36):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
24	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_35):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
23	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_34):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
22	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_33):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_32):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
20	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_31):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
19	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_175):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
18	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_174):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
17	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_173):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_172):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
15	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_171):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
14	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_170):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_169):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_168):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_167):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_166):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_165):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_164):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_163):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_162):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_161):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_160):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_159):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_158):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_157):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_156):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.46 Reserved (RSVD4[0])—Offset B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.47 Reserved (RSVD4[1])—Offset B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.48 Reserved (RSVD4[2])—Offset BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.49 Reserved (RSVD4[3])—Offset C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.50 Reserved (RSVD4[4])—Offset C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.51 Reserved (RSVD4[5])—Offset C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.52 Reserved (RSVD4[6])—Offset CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.53 Reserved (RSVD4[7])—Offset D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.54 Reserved (RSVD4[8])—Offset D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.55 Reserved (RSVD4[9])—Offset D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.56 Reserved (RSVD4[10])—Offset DCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.57 Reserved (RSVD4[11])—Offset E0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.58 Reserved (RSVD4[12])—Offset E4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.59 Reserved (RSVD4[13])—Offset E8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.60 Reserved (RSVD4[14])—Offset ECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.61 Reserved (RSVD4[15])—Offset F0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.62 Reserved (RSVD4[16])—Offset F4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.63 Reserved (RSVD4[17])—Offset F8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.64 Reserved (RSVD4[18])—Offset FCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.65 GPI Interrupt Status (GPI_IS_audio_0)—Offset 100h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD_0):** Reserved
27	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_38):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_37):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
25	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_36):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
24	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_35):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
23	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_34):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
22	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_33):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
21	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_32):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
20	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_31):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
19	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_175):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_174):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
17	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_173):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
16	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_172):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
15	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_171):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_170):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_169):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_168):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_167):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_166):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_165):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_164):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_163):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_162):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_161):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_160):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_159):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_158):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_157):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_156):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.66 Reserved (RSVD5[0])—Offset 104h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.67 Reserved (RSVD5[1])—Offset 108h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.68 Reserved (RSVD5[2])—Offset 10Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.69 GPI Interrupt Enable (GPI_IE_audio_0)—Offset 110h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD_0):** Reserved
27	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_38):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_37):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
25	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_36):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
24	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_35):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
23	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_34):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
22	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_33):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
21	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_32):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_31):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
19	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_175):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
18	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_174):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
17	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_173):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
16	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_172):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
15	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_171):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_170):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_169):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_168):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_167):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_166):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_165):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_164):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_163):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
6	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_162):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_161):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_160):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_159):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_158):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_157):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_156):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.70 Reserved (RSVD6[0])—Offset 114h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.71 Reserved (RSVD6[1])—Offset 118h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.72 Reserved (RSVD6[2])—Offset 11Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.73 Reserved (RSVD6[3])—Offset 120h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.74 Reserved (RSVD6[4])—Offset 124h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.75 Reserved (RSVD6[5])—Offset 128h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.76 Reserved (RSVD6[6])—Offset 12Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.77 GPI General Purpose Events Status (GPI_GPE_STS_audio_0)—Offset 130h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD_0):** Reserved
27	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_38):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
26	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_37):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
25	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_36):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
24	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_35):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
23	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_34):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
22	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_33):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
21	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_32):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
20	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_31):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
19	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_175):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_174):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
17	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_173):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
16	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_172):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
15	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_171):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_170):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_169):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_168):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_167):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_166):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
9	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_165):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_164):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_163):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_162):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
5	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_161):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_160):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_159):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_158):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_157):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_156):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.78 Reserved (RSVD7[0])—Offset 134h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.79 Reserved (RSVD7[1])—Offset 138h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.80 Reserved (RSVD7[2])—Offset 13Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.81 Reserved (RSVD7[3])—Offset 140h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.82 Reserved (RSVD7[4])—Offset 144h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.83 Reserved (RSVD7[5])—Offset 148h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.84 Reserved (RSVD7[6])—Offset 14Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.85 GPI General Purpose Events Enable (GPI_GPE_EN_audio_0)—Offset 150h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_38):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
26	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_37):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
25	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_36):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
24	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_35):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
23	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_34):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
22	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_33):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
21	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_32):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
20	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_31):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
19	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_175):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
18	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_174):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_173):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
16	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_172):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
15	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_171):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
14	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_170):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_169):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_168):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_167):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_166):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
9	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_165):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_164):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_163):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_162):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
5	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_161):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_160):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_159):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_158):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_157):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_156):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.86 Reserved (RSVD8[0])—Offset 154h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.87 Reserved (RSVD8[1])—Offset 158h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.88 Reserved (RSVD8[2])—Offset 15Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.89 Reserved (RSVD8[3])—Offset 160h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.90 Reserved (RSVD8[4])—Offset 164h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.91 Reserved (RSVD8[5])—Offset 168h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.92 Reserved (RSVD8[6])—Offset 16Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.93 SMI Status (GPI_SMI_STS_audio_0)—Offset 170h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_38):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
26	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_37):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
25	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_36):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
24	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_35):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
23	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_34):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
22	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_33):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_32):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
20	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_31):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
19	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_175):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_174):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
17	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_173):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
16	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_172):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_171):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
14	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_170):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
13	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_169):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_168):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
11	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_167):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
10	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_166):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_165):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
8	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_164):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
7	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_163):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_162):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
5	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_161):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
4	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_160):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_159):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
2	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_158):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
1	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_157):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_156):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

29.94 Reserved (RSVD9[0])—Offset 174h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.95 Reserved (RSVD9[1])—Offset 178h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.96 Reserved (RSVD9[2])—Offset 17Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.97 Reserved (RSVD9[3])—Offset 180h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.98 Reserved (RSVD9[4])—Offset 184h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.99 Reserved (RSVD9[5])—Offset 188h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.100 Reserved (RSVD9[6])—Offset 18Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.101 SMI Enable (GPI_SMI_EN_audio_0)—Offset 190h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD_0):** Reserved
27	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_38):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
26	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_37):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
25	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_36):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
24	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_35):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
23	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_34):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
22	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_33):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
21	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_32):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
20	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_31):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
19	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_175):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
18	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_174):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_173):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
16	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_172):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
15	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_171):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
14	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_170):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_169):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
12	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_168):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
11	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_167):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
10	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_166):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_165):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
8	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_164):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
7	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_163):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
6	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_162):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_161):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
4	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_160):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
3	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_159):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
2	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_158):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_157):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
0	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_156):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

29.102 Reserved (RSVD10[0])—Offset 194h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.103 Reserved (RSVD10[1])—Offset 198h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.104 Reserved (RSVD10[2])—Offset 19Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.105 Reserved (RSVD10[3])—Offset 1A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.106 Reserved (RSVD10[4])—Offset 1A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.107 Reserved (RSVD10[5])—Offset 1A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.108 Reserved (RSVD10[6])—Offset 1ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.109 NMI Status (GPI_NMI_STS_audio_0)—Offset 1B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD_0):** Reserved
27	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_38):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
26	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_37):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
25	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_36):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
24	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_35):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
23	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_34):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
22	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_33):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
21	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_32):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
20	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_31):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
19	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_175):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_174):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
17	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_173):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
16	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_172):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
15	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_171):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
14	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_170):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
13	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_169):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
12	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_168):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
11	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_167):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_166):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
9	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_165):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
8	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_164):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
7	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_163):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_162):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
5	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_161):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
4	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_160):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
3	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_159):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_158):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
1	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_157):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
0	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_156):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.110 Reserved (RSVD11[0])—Offset 1B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.111 Reserved (RSVD11[1])—Offset 1B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.112 Reserved (RSVD11[2])—Offset 1BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.113 Reserved (RSVD11[3])—Offset 1C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.114 Reserved (RSVD11[4])—Offset 1C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.115 Reserved (RSVD11[5])—Offset 1C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.116 Reserved (RSVD11[6])—Offset 1CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.117 NMI Enable (GPI_NMI_EN_audio_0)—Offset 1D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_38):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
26	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_37):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
25	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_36):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
24	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_35):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_34):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
22	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_33):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
21	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_32):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
20	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_31):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_175):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
18	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_174):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
17	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_173):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
16	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_172):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_171):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
14	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_170):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
13	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_169):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
12	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_168):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_167):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
10	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_166):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
9	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_165):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
8	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_164):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_163):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
6	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_162):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
5	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_161):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
4	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_160):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_159):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
2	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_158):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
1	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_157):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
0	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_156):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.118 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_AVS_1P8)—Offset 300h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.119 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_AVS_1P8)—Offset 304h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.120 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_AVS_1P8)—Offset 308h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RO	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.121 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_AVS_1P8)—Offset 30Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RO	**strsel B (STRSELB):** Strsel B value
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.122 Family Configuration Register (FAM_CFG_AVS_1P8)—Offset 310h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1C00000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
26	0h RO	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	0h RW	**strsel (STRSEL):** Pad driver impedance selection

Bit Range	Default & Access	Field Name (ID): Description
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RO	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.123 Family Reserved Register DW5 (FAM_RSVD_DW5AVS_1P8)—Offset 314h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.124 Family Reserved Register DW6 (FAM_RSVD_DW6AVS_1P8)—Offset 318h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.125 Family Reserved Register DW7 (FAM_RSVD_DW7AVS_1P8)—Offset 31Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.126 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_v_CNV1)—Offset 320h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RO	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.127 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_v_CNV1)—Offset 324h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.128 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_v_CNV1)—Offset 328h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RO	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.129 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_v_CNV1)—Offset 32Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RO	**strsel B (STRSELB):** Strsel B value
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.130 Family Configuration Register (FAM_CFG_v_CNV1)—Offset 330h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C00000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RO	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
26	0h RO	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	0h RO	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	0h RO	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RO	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.131 Family Reserved Register DW5 (FAM_RSVD_DW5v_CNV1)—Offset 334h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.132 Family Reserved Register DW6 (FAM_RSVD_DW6v_CNV1)—Offset 338h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.133 Family Reserved Register DW7 (FAM_RSVD_DW7v_CNV1)—Offset 33Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.134 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_156)—Offset 600h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).

Bit Range	Default & Access	Field Name (ID): Description
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.135 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_156)—Offset 604h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.136 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_156)—Offset 608h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.137 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_156)—Offset 60Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.138 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_157)—Offset 610h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.139 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_157)—Offset 614h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.140 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_157)—Offset 618h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.141 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_157)—Offset 61Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.142 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_158)—Offset 620h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.143 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_158)—Offset 624h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.144 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_158)—Offset 628h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.145 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_158)—Offset 62Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.146 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_159)—Offset 630h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.147 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_159)—Offset 634h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.148 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_159)—Offset 638h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.149 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_159)—Offset 63Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.150 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_160)—Offset 640h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.151 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_160)—Offset 644h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.152 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_160)—Offset 648h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.153 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_160)—Offset 64Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.154 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_161)—Offset 650h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.155 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_161)—Offset 654h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.156 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_161)—Offset 658h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.157 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_161)—Offset 65Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.158 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_162)—Offset 660h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.159 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_162)—Offset 664h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.160 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_162)—Offset 668h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.161 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_162)—Offset 66Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.162 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_163)—Offset 670h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.163 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_163)—Offset 674h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.164 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_163)—Offset 678h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 800h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	1h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.165 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_163)—Offset 67Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.166 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_164)—Offset 680h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.167 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_164)—Offset 684h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.168 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_164)—Offset 688h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.169 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_164)—Offset 68Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.170 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_165)—Offset 690h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.171 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_165)—Offset 694h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.172 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_165)—Offset 698h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.173 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_165)—Offset 69Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.174 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_166)—Offset 6A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.175 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_166)—Offset 6A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.176 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_166)—Offset 6A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.177 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_166)—Offset 6ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.178 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_167)—Offset 6B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.179 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_167)—Offset 6B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.180 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_167)—Offset 6B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.181 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_167)—Offset 6BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.182 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_168)—Offset 6C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.183 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_168)—Offset 6C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.184 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_168)—Offset 6C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 800h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	1h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.185 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_168)—Offset 6CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.186 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_169)—Offset 6D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.187 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_169)—Offset 6D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.188 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_169)—Offset 6D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.189 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_169)—Offset 6DCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.190 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_170)—Offset 6E0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.191 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_170)—Offset 6E4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.192 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_170)—Offset 6E8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.193 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_170)—Offset 6ECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.194 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_171)—Offset 6F0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.195 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_171)—Offset 6F4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.196 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_171)—Offset 6F8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.197 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_171)—Offset 6FCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.198 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_172)—Offset 700h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.199 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_172)—Offset 704h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.200 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_172)—Offset 708h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.201 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_172)—Offset 70Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.202 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_173)—Offset 710h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.203 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_173)—Offset 714h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.204 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_173)—Offset 718h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.205 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_173)—Offset 71Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.206 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_174)—Offset 720h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.207 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_174)—Offset 724h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.208 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_174)—Offset 728h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.209 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_174)—Offset 72Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.210 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_175)—Offset 730h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.211 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_175)—Offset 734h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.212 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_175)—Offset 738h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.213 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_175)—Offset 73Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.214 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_31)—Offset 740h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.215 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_31)—Offset 744h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.216 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_31)—Offset 748h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.217 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_31)—Offset 74Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.218 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_32)—Offset 750h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.219 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_32)—Offset 754h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.220 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_32)—Offset 758h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.221 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_32)—Offset 75Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.222 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_33)—Offset 760h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.223 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_33)—Offset 764h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.224 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_33)—Offset 768h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.225 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_33)—Offset 76Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.226 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_34)—Offset 770h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.227 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_34)—Offset 774h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.228 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_34)—Offset 778h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.229 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_34)—Offset 77Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.230 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_35)—Offset 780h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.231 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_35)—Offset 784h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.232 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_35)—Offset 788h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.233 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_35)—Offset 78Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.234 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_36)—Offset 790h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.235 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_36)—Offset 794h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.236 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_36)—Offset 798h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.237 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_36)—Offset 79Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.238 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_37)—Offset 7A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.239 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_37)—Offset 7A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.240 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_37)—Offset 7A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.241 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_37)—Offset 7ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.242 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_38)—Offset 7B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.243 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_38)—Offset 7B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.244 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_38)—Offset 7B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.245 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_38)—Offset 7BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.246 Revision ID (REV_ID)—Offset 0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 940000h

Bit Range	Default & Access	Field Name (ID): Description
31:16	94h RO	**Hardware Base Revision ID (HWBASEREVID):** This field is used to indicate the baseline hardware revision. 0091h = RTL release for HAS0.91
15:0	0h RO	**Hardware Sub Revision ID (HWSUBIREVID):** This field is used to indicate the revision of the hardware derived from the baseline.

29.247 Capability List Register (CAP_LIST_0)—Offset 4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD_0):** Reserved
23:16	0h RO	**Capability Identification (CAPID):** A unique identification for the current capability. A value of 0 is reserved, and must not be used by any capability. Note:This field is always 0 for the first Capability List register.
15:0	0h RO	**Next Capability List Pointer (NXTCAPLPTR):** Specify the DW-aligned Pointer/ Address to the next item in this capabilities list and must be 0 if there is no capability at all or this is the last capability in the list.

29.248 Family Base Address (FAMBAR)—Offset 8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 300h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15:0	300h RO	**Family Base Address (FAMBAR):** This field provides the starting byte-align address of Family0 register sets within a Community. It is meant for software to discover from where the very first Family register (i.e. Family0 register) starts to compute the next Families address offsets.

29.249 Pad Base Address (PADBAR)—Offset Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 600h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
15:0	600h RO	**Pad Base Address (PADBAR):** This field provides the starting byte-align address of Pad0 register sets within a Community. It is meant for software to discover from where the very first Pad register (i.e. Pad0 register) starts to compute the next Pad address offsets.

29.250 Miscellaneous Configuration (MISCCFG)—Offset 10h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: E043200h

Bit Range	Default & Access	Field Name (ID): Description
31:24	Eh RW	**GPIO Driver Mode Interrupt Select (GPDMINTSEL):** IRQ globally for all pads (GPI_IS with corresponding GPI_IE enable). 0 = Interrupt Line 0 1 = Interrupt Line 1 ... 255 = Interrupt Line 255
23:20	0h RW	**Reserved (Reserved1):** Reserved
19:16	4h RW	**GPIO Group to GPE_DW2 assignment encoding (GPE0_DW2):** This register assigns a specific GPIO Group to the ACPI GPE0[95:64]. The setting is SoC specific.
15:12	3h RW	**GPIO Group to GPE_DW1 assignment encoding (GPE0_DW1):** This register assigns a specific GPIO Group to the ACPI GPE0[63:32]. The setting is SoC specific.
11:8	2h RW	**GPIO Group to GPE_DW0 assignment encoding (GPE0_DW0):** This register assigns a specific GPIO Group to the ACPI GPE0[31:0]. The setting is SoC specific.
7:6	0h RW	**Reserved (Reserved2):** Reserved
5	0h RW	**GPIO IOSF Sideband Dynamic Partition Clock Gating Enable (GPSIDEDPCGEN):** Specify whether the GPIO Community IOSF sideband clock should take part in partition clock gating 0 = Disable participation in IOSF sideband clock dynamic partition clock gating 1 = Enable participation in IOSF sideband clock dynamic partition clock gating
4	0h RW	**GPIO RCOMP clock Dynamic Local Clock Gating (GPRCOMPCDLCGEN):** Specify whether the GPIO Community should perform local clock gating on RCOMP clock 0 = Disable dynamic RCOMP clock local clock gating 1 = Enable dynamic RCOMP clock local clock gating
3	0h RW	**GPIO RTC clock Dynamic Local Clock Gating (GPRTCDLCGEN):** Specify whether the GPIO Community should perform local clock gating on RTC clock 0 = Disable dynamic RTC clock local clock gating 1 = Enable dynamic RTC clock local clock gating
2	0h RW	**GSX Static Local Clock Gating (GSXSLCGEN):** Specify whether the GSX controller should be statically clock gated for power saving if it is not enabled (even though the capability is available). 0 = Disable static local clock gating 1 = Enable static local clock gating
1	0h RW	**GPIO Dynamic Partition Clock Gating Enable (GPDPCGEN):** Specify whether the GPIO Community should take part in partition clock gating 0 = Disable participation in dynamic partition clock gating 1 = Enable participation in dynamic partition clock gating
0	0h RW	**GPIO Dynamic Local Clock Gating Enable (GPDLCGEN):** Specify whether the GPIO Community should perform local clock gating 0 = Disable dynamic local clock gating 1 = Enable dynamic local clock gating

29.251 Miscellaneous Secured Configuration (MISCSECCFG)—Offset 14h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD_0):** Reserved
7:4	0h RW	**Reserved (Reserved1):** Reserved
3	0h RW/1C/V	**Soft strap pull status bit (SSTRAPPULLSTAT):** Indicate the status of the last soft strap pull operation. If the implementation guarantee the success of the operation, then this bit can be hard-wired to 0. 0 = No error 1 = Error FW writes 1 to this bit to clear. This bit is set to 1 when soft strap pull request has been completed with completion without data (for both successful and unsuccessful completion status) or completion with data but with unsuccessful completion status. This bit will also set to 1 if there is any failure on the security and error checking in the fuse puller such as invalid SAI and etc. Fuse puller will assert puller_error in this case and this will set this bit to 1.
2	0h RW/1C/V	**Soft strap re-pull done bit (SSTRAPPULLDONE):** Indication if the soft strap pull has finished the operation 0 = operation is not done. Either not started or in progress 1 = operation is done/finish. FW writes 1 to this bit to clear.
1	0h RW	**Soft strap re-pull request bit (SSTRAPREPULLREQ):** Specify if fuse puller need to re-pull the soft strap. The SoC/PMC/FW can set this bit to request the community fuse puller to re-pull the soft strap. 0 to 1 transition = soft strap re-pull request trigger. A soft strap re-pull request will not trigger IP_READY message sent as the IP_READY message shall be sent during the initial soft strap pull by HW as part of the reset sequences. The SSTRAPREPULLREQ, SSTRAPPULLDONE and SSTRAPPULLSTAT are only meaningful for community with soft strap pull capability.
0	0h RW	**Event Trigger IOSF-SB Message Initiation Disable (SBTRIGDIS):** This bit disables the following list of IOSF-SB Message initiation. It does not disable the event trigger. When this bit is cleared to '0', any pending message due to event trigger while this bit is '1' shall be sent on IOSF-SB. Event list: 'Virtual Wire' message IRQ GPE/SCI NMI SMI

29.252 Reserved (RSVD0[0])—Offset 18h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.253 Reserved (RSVD0[1])—Offset 1Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.254 Pad Ownership (PAD_OWN_north_0)—Offset 20h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_83):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_82):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_81):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_80):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_79):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_78):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_77):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_76):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.255 Pad Ownership (PAD_OWN_north_1)—Offset 24h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_91):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_90):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_89):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_88):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_87):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_86):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_85):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_84):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.256 Pad Ownership (PAD_OWN_north_2)—Offset 28h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_99):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_98):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_97):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_96):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_95):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_94):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_93):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_92):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.257 Pad Ownership (PAD_OWN_north_3)—Offset 2Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_107):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_106):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_105):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_104):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_103):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_102):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_101):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_100):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.258 Pad Ownership (PAD_OWN_north_4)—Offset 30h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_115):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_114):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_113):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_112):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_111):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_110):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_109):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_108):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.259 Pad Ownership (PAD_OWN_north_5)—Offset 34h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_123):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_122):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_121):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_120):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_119):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_118):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_117):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_116):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.260 Pad Ownership (PAD_OWN_north_6)—Offset 38h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_131):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_130):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_129):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_128):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_127):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_126):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_125):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_124):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.261 Pad Ownership (PAD_OWN_north_7)—Offset 3Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_139):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_138):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_137):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_136):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_135):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_134):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_133):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_132):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.262 Pad Ownership (PAD_OWN_north_8)—Offset 40h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_147):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_146):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_145):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_144):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_143):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_142):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_141):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_140):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.263 Pad Ownership (PAD_OWN_north_9)—Offset 44h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_155):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_154):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_153):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_152):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_151):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_150):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_149):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_148):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.264 Reserved (RSVD1[0])—Offset 48h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.265 Reserved (RSVD1[1])—Offset 4Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.266 Reserved (RSVD1[2])—Offset 50h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.267 Reserved (RSVD1[3])—Offset 54h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.268 Reserved (RSVD1[4])—Offset 58h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.269 Reserved (RSVD1[5])—Offset 5Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.270 GPI Virtual Wire Message Enable (GPI_VWE_north_0)—Offset 60h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_107):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
30	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_106):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
29	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_105):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
28	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_104):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_103):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
26	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_102):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
25	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_101):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
24	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_100):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
23	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_99):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
22	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_98):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
21	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_97):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
20	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_96):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
19	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_95):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
18	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_94):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
17	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_93):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
16	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_92):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_91):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
14	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_90):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_89):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_88):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
11	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_87):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_86):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
9	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_85):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_84):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_83):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_82):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
5	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_81):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_80):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_79):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_78):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_77):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_76):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.271 GPI Virtual Wire Message Enable (GPI_VWE_north_1)—Offset 64h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_139):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
30	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_138):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
29	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_137):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
28	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_136):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
27	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_135):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
26	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_134):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
25	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_133):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
24	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_132):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
23	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_131):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
22	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_130):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
21	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_129):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_128):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
19	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_127):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
18	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_126):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
17	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_125):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
16	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_124):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
15	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_123):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
14	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_122):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_121):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
12	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_120):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_119):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_118):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
9	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_117):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
8	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_116):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_115):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_114):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
5	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_113):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
4	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_112):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_111):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_110):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_109):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_108):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.272 GPI Virtual Wire Message Enable (GPI_VWE_north_2)—Offset 68h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_155):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
14	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_154):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_153):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_152):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_151):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
10	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_150):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
9	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_149):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_148):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_147):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
6	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_146):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
5	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_145):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_144):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_143):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
2	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_142):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_141):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_140):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.273 Reserved (RSVD2[0])—Offset 6Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.274 Reserved (RSVD2[1])—Offset 70h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.275 Reserved (RSVD2[2])—Offset 74h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.276 Reserved (RSVD2[3])—Offset 78h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.277 Reserved (RSVD2[4])—Offset 7Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.278 Pad Configuration Lock (PADCFGLOCK_north_0)—Offset 80h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_107):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
30	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_106):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
29	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_105):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
28	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_104):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
27	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_103):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_102):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
25	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_101):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
24	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_100):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
23	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_99):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
22	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_98):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
21	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_97):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_96):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
19	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_95):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_94):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
17	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_93):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_92):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
15	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_91):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_90):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
13	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_89):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_88):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
11	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_87):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_86):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
9	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_85):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_84):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
7	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_83):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_82):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
5	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_81):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_80):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
3	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_79):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_78):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
1	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_77):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_76):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.279 Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_north_0)—Offset 84h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_107):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
30	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_106):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
29	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_105):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
28	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_104):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_103):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
26	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_102):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_101):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
24	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_100):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_99):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
22	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_98):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_97):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
20	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_96):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_95):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
18	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_94):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_93):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
16	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_92):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_91):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
14	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_90):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_89):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
12	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_88):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_87):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
10	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_86):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_85):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
8	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_84):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_83):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
6	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_82):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_81):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
4	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_80):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_79):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
2	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_78):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_77):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
0	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_76):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.280 Pad Configuration Lock (PADCFGLOCK_north_1)—Offset 88h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_139):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
30	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_138):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
29	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_137):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
28	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_136):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_135):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
26	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_134):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_133):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
24	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_132):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_131):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
22	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_130):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_129):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
20	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_128):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_127):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
18	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_126):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_125):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
16	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_124):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_123):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
14	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_122):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_121):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
12	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_120):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_119):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
10	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_118):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_117):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
8	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_116):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_115):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
6	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_114):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_113):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
4	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_112):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_111):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
2	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_110):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_109):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
0	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_108):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.281 Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_north_1)—Offset 8Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_139):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
30	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_138):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
29	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_137):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
28	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_136):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_135):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
26	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_134):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_133):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
24	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_132):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_131):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
22	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_130):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_129):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
20	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_128):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_127):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
18	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_126):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_125):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
16	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_124):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_123):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
14	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_122):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_121):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
12	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_120):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_119):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
10	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_118):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_117):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
8	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_116):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_115):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
6	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_114):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_113):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
4	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_112):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_111):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
2	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_110):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_109):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
0	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_108):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.282 Pad Configuration Lock (PADCFGLOCK_north_2)—Offset 90h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_155):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
14	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_154):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_153):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
12	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_152):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_151):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
10	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_150):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_149):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
8	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_148):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_147):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
6	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_146):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_145):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
4	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_144):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_143):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
2	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_142):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_141):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
0	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_140):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.283 Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_north_2)—Offset 94h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_155):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
14	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_154):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_153):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
12	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_152):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_151):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
10	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_150):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_149):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
8	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_148):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_147):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
6	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_146):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_145):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
4	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_144):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_143):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
2	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_142):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_141):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
0	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_140):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.284 Reserved (RSVD3[0])—Offset 98h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.285 Reserved (RSVD3[1])—Offset 9Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.286 Reserved (RSVD3[2])—Offset A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.287 Reserved (RSVD3[3])—Offset A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.288 Reserved (RSVD3[4])—Offset A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.289 Reserved (RSVD3[5])—Offset ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.290 Host Software Pad Ownership (HOSTSW_OWN_north_0)—Offset B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_107):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
30	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_106):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
29	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_105):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
28	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_104):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
27	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_103):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
26	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_102):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
25	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_101):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
24	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_100):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
23	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_99):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
22	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_98):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
21	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_97):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
20	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_96):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_95):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
18	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_94):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
17	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_93):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
16	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_92):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
15	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_91):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_90):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_89):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_88):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_87):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_86):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_85):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_84):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_83):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
6	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_82):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_81):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_80):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_79):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_78):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_77):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_76):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.291 Host Software Pad Ownership (HOSTSW_OWN_north_1)—Offset B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_139):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
30	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_138):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
29	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_137):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
28	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_136):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
27	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_135):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_134):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
25	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_133):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
24	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_132):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
23	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_131):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
22	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_130):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_129):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
20	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_128):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
19	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_127):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
18	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_126):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
17	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_125):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_124):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
15	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_123):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
14	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_122):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_121):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_120):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_119):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_118):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_117):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_116):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_115):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_114):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_113):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_112):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_111):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_110):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_109):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_108):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.292 Host Software Pad Ownership (HOSTSW_OWN_north_2)—Offset B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_155):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_154):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_153):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_152):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_151):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_150):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_149):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_148):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_147):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
6	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_146):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_145):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_144):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_143):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_142):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_141):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_140):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.293 Reserved (RSVD4[0])—Offset BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.294 Reserved (RSVD4[1])—Offset C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.295 Reserved (RSVD4[2])—Offset C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.296 Reserved (RSVD4[3])—Offset C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.297 Reserved (RSVD4[4])—Offset CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.298 Reserved (RSVD4[5])—Offset D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.299 Reserved (RSVD4[6])—Offset D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.300 Reserved (RSVD4[7])—Offset D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.301 Reserved (RSVD4[8])—Offset DCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.302 Reserved (RSVD4[9])—Offset E0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.303 Reserved (RSVD4[10])—Offset E4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.304 Reserved (RSVD4[11])—Offset E8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.305 Reserved (RSVD4[12])—Offset ECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.306 Reserved (RSVD4[13])—Offset F0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.307 Reserved (RSVD4[14])—Offset F4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.308 Reserved (RSVD4[15])—Offset F8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.309 Reserved (RSVD4[16])—Offset FCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.310 GPI Interrupt Status (GPI_IS_north_0)—Offset 100h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_107):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
30	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_106):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
29	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_105):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
28	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_104):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
27	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_103):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_102):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
25	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_101):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
24	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_100):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
23	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_99):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
22	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_98):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
21	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_97):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
20	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_96):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
19	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_95):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_94):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
17	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_93):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
16	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_92):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
15	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_91):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_90):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_89):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_88):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_87):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_86):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_85):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_84):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_83):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_82):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_81):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_80):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_79):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_78):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_77):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_76):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.311 GPI Interrupt Status (GPI_IS_north_1)—Offset 104h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_139):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
30	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_138):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
29	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_137):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
28	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_136):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_135):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
26	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_134):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
25	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_133):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
24	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_132):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_131):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
22	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_130):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
21	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_129):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
20	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_128):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_127):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
18	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_126):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
17	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_125):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
16	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_124):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_123):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
14	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_122):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_121):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_120):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_119):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_118):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_117):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_116):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_115):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
6	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_114):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_113):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_112):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_111):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_110):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_109):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_108):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.312 GPI Interrupt Status (GPI_IS_north_2)—Offset 108h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_155):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
14	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_154):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_153):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_152):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_151):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_150):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_149):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_148):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_147):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
6	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_146):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_145):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_144):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_143):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_142):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_141):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_140):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.313 Reserved (RSVD5)—Offset 10Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.314 GPI Interrupt Enable (GPI_IE_north_0)—Offset 110h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_107):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
30	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_106):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
29	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_105):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
28	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_104):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
27	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_103):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
26	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_102):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
25	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_101):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
24	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_100):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
23	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_99):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
22	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_98):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
21	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_97):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
20	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_96):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
19	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_95):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_94):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
17	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_93):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
16	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_92):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
15	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_91):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
14	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_90):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_89):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_88):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_87):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_86):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_85):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_84):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_83):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_82):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_81):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_80):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_79):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_78):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_77):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_76):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.315 GPI Interrupt Enable (GPI_IE_north_1)—Offset 114h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_139):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
30	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_138):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
29	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_137):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
28	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_136):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
27	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_135):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
26	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_134):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
25	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_133):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
24	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_132):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
23	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_131):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
22	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_130):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
21	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_129):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
20	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_128):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
19	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_127):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
18	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_126):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
17	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_125):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_124):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
15	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_123):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
14	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_122):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_121):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_120):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_119):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_118):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_117):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_116):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_115):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
6	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_114):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_113):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_112):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_111):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_110):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_109):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_108):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.316 GPI Interrupt Enable (GPI_IE_north_2)—Offset 118h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_155):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
14	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_154):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_153):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_152):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_151):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_150):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_149):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_148):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_147):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
6	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_146):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_145):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_144):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_143):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_142):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_141):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_140):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.317 Reserved (RSVD6[0])—Offset 11Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.318 Reserved (RSVD6[1])—Offset 120h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.319 Reserved (RSVD6[2])—Offset 124h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.320 Reserved (RSVD6[3])—Offset 128h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.321 Reserved (RSVD6[4])—Offset 12Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.322 GPI General Purpose Events Status (GPI_GPE_STS_north_0)—Offset 130h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_107):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
30	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_106):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
29	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_105):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
28	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_104):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
27	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_103):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
26	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_102):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
25	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_101):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
24	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_100):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
23	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_99):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
22	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_98):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
21	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_97):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
20	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_96):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
19	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_95):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
18	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_94):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_93):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
16	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_92):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
15	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_91):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
14	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_90):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_89):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_88):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_87):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_86):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_85):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_84):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_83):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_82):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_81):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_80):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_79):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_78):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_77):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_76):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.323 GPI General Purpose Events Status (GPI_GPE_STS_north_1)—Offset 134h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_139):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
30	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_138):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
29	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_137):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
28	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_136):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
27	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_135):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_134):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
25	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_133):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
24	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_132):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
23	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_131):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
22	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_130):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
21	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_129):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
20	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_128):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
19	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_127):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_126):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
17	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_125):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
16	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_124):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
15	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_123):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_122):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_121):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_120):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_119):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_118):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
9	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_117):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_116):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_115):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_114):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
5	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_113):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_112):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_111):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_110):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_109):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_108):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.324 GPI General Purpose Events Status (GPI_GPE_STS_north_2)—Offset 138h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_155):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
14	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_154):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_153):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_152):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_151):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_150):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
9	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_149):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_148):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_147):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_146):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
5	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_145):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_144):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_143):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_142):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_141):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_140):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.325 Reserved (RSVD7[0])—Offset 13Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.326 Reserved (RSVD7[1])—Offset 140h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.327 Reserved (RSVD7[2])—Offset 144h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.328 Reserved (RSVD7[3])—Offset 148h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.329 Reserved (RSVD7[4])—Offset 14Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.330 GPI General Purpose Events Enable (GPI_GPE_EN_north_0)—Offset 150h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_107):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
30	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_106):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
29	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_105):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
28	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_104):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
27	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_103):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
26	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_102):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
25	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_101):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
24	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_100):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
23	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_99):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
22	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_98):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
21	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_97):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
20	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_96):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_95):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
18	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_94):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
17	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_93):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
16	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_92):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
15	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_91):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_90):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_89):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_88):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_87):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_86):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_85):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_84):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_83):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_82):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
5	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_81):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_80):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_79):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_78):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_77):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_76):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.331 GPI General Purpose Events Enable (GPI_GPE_EN_north_1)—Offset 154h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_139):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
30	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_138):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
29	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_137):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
28	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_136):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_135):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
26	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_134):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
25	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_133):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
24	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_132):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
23	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_131):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
22	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_130):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
21	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_129):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
20	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_128):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
19	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_127):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
18	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_126):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_125):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
16	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_124):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
15	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_123):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
14	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_122):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_121):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_120):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_119):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_118):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
9	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_117):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_116):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_115):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_114):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
5	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_113):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_112):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_111):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_110):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_109):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_108):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.332 GPI General Purpose Events Enable (GPI_GPE_EN_north_2)—Offset 158h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_155):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
14	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_154):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_153):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_152):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_151):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_150):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
9	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_149):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_148):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_147):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_146):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_145):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_144):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_143):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_142):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_141):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_140):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.333 Reserved (RSVD8[0])—Offset 15Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.334 Reserved (RSVD8[1])—Offset 160h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.335 Reserved (RSVD8[2])—Offset 164h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.336 Reserved (RSVD8[3])—Offset 168h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.337 Reserved (RSVD8[4])—Offset 16Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.338 SMI Status (GPI_SMI_STS_north_0)—Offset 170h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_107):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
30	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_106):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
29	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_105):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
28	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_104):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
27	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_103):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
26	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_102):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_101):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
24	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_100):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
23	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_99):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
22	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_98):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
21	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_97):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
20	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_96):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_95):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
18	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_94):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
17	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_93):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
16	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_92):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
15	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_91):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
14	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_90):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_89):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
12	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_88):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
11	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_87):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_86):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
9	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_85):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
8	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_84):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_83):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
6	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_82):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
5	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_81):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_80):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
3	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_79):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
2	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_78):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_77):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
0	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_76):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

29.339 SMI Status (GPI_SMI_STS_north_1)—Offset 174h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_139):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
30	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_138):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
29	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_137):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
28	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_136):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
27	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_135):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
26	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_134):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_133):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
24	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_132):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
23	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_131):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
22	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_130):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
21	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_129):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
20	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_128):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_127):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
18	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_126):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
17	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_125):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_124):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
15	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_123):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
14	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_122):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_121):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
12	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_120):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
11	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_119):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_118):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
9	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_117):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
8	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_116):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_115):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
6	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_114):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
5	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_113):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_112):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
3	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_111):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
2	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_110):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_109):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
0	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_108):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

29.340 SMI Status (GPI_SMI_STS_north_2)—Offset 178h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_155):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
14	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_154):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
13	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_153):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_152):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
11	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_151):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
10	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_150):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_149):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
8	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_148):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
7	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_147):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_146):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
5	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_145):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
4	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_144):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_143):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
2	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_142):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
1	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_141):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_140):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

29.341 Reserved (RSVD9[0])—Offset 17Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.342 Reserved (RSVD9[1])—Offset 180h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.343 Reserved (RSVD9[2])—Offset 184h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.344 Reserved (RSVD9[3])—Offset 188h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.345 Reserved (RSVD9[4])—Offset 18Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.346 SMI Enable (GPI_SMI_EN_north_0)—Offset 190h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_107):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
30	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_106):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
29	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_105):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
28	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_104):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_103):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
26	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_102):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
25	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_101):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
24	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_100):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
23	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_99):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
22	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_98):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
21	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_97):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
20	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_96):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
19	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_95):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
18	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_94):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
17	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_93):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
16	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_92):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_91):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
14	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_90):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
13	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_89):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
12	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_88):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_87):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
10	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_86):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
9	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_85):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
8	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_84):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_83):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
6	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_82):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
5	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_81):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
4	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_80):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_79):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
2	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_78):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
1	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_77):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
0	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_76):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

29.347 SMI Enable (GPI_SMI_EN_north_1)—Offset 194h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_139):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
30	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_138):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
29	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_137):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
28	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_136):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_135):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
26	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_134):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
25	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_133):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
24	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_132):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_131):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
22	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_130):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
21	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_129):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
20	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_128):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_127):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
18	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_126):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
17	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_125):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
16	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_124):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_123):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
14	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_122):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
13	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_121):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
12	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_120):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_119):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
10	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_118):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
9	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_117):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
8	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_116):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_115):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
6	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_114):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
5	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_113):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
4	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_112):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_111):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
2	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_110):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
1	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_109):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
0	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_108):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

29.348 SMI Enable (GPI_SMI_EN_north_2)—Offset 198h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_155):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
14	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_154):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
13	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_153):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_152):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
11	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_151):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
10	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_150):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
9	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_149):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_148):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
7	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_147):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
6	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_146):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
5	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_145):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_144):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
3	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_143):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
2	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_142):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
1	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_141):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_140):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

29.349 Reserved (RSVD10[0])—Offset 19Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.350 Reserved (RSVD10[1])—Offset 1A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.351 Reserved (RSVD10[2])—Offset 1A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.352 Reserved (RSVD10[3])—Offset 1A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.353 Reserved (RSVD10[4])—Offset 1ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.354 NMI Status (GPI_NMI_STS_north_0)—Offset 1B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_107):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
30	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_106):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
29	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_105):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
28	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_104):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_103):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
26	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_102):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
25	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_101):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
24	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_100):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_99):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
22	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_98):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
21	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_97):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
20	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_96):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_95):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
18	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_94):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
17	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_93):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
16	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_92):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_91):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
14	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_90):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
13	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_89):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
12	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_88):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_87):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
10	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_86):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
9	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_85):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
8	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_84):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_83):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
6	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_82):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
5	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_81):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
4	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_80):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_79):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
2	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_78):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
1	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_77):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
0	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_76):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.355 NMI Status (GPI_NMI_STS_north_1)—Offset 1B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_139):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
30	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_138):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
29	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_137):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
28	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_136):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
27	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_135):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
26	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_134):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
25	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_133):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
24	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_132):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
23	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_131):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
22	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_130):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
21	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_129):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_128):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
19	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_127):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
18	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_126):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
17	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_125):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
16	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_124):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
15	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_123):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
14	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_122):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
13	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_121):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_120):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
11	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_119):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
10	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_118):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
9	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_117):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
8	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_116):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
7	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_115):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
6	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_114):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
5	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_113):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_112):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
3	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_111):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
2	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_110):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
1	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_109):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_108):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.356 NMI Status (GPI_NMI_STS_north_2)—Offset 1B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_155):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
14	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_154):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_153):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
12	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_152):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
11	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_151):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
10	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_150):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_149):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
8	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_148):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
7	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_147):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
6	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_146):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_145):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
4	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_144):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
3	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_143):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
2	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_142):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_141):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
0	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_140):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.357 Reserved (RSVD11[0])—Offset 1BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.358 Reserved (RSVD11[1])—Offset 1C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.359 Reserved (RSVD11[2])—Offset 1C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.360 Reserved (RSVD11[3])—Offset 1C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.361 Reserved (RSVD11[4])—Offset 1CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.362 NMI Enable (GPI_NMI_EN_north_0)—Offset 1D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_107):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
30	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_106):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
29	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_105):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
28	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_104):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_103):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
26	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_102):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
25	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_101):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
24	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_100):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_99):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
22	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_98):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
21	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_97):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
20	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_96):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_95):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
18	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_94):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
17	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_93):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
16	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_92):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_91):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
14	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_90):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
13	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_89):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
12	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_88):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_87):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
10	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_86):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
9	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_85):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
8	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_84):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_83):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
6	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_82):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
5	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_81):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
4	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_80):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_79):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
2	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_78):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
1	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_77):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
0	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_76):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.363 NMI Enable (GPI_NMI_EN_north_1)—Offset 1D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_139):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
30	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_138):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
29	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_137):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
28	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_136):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_135):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
26	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_134):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
25	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_133):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
24	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_132):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_131):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
22	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_130):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
21	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_129):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
20	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_128):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_127):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
18	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_126):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
17	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_125):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
16	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_124):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_123):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
14	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_122):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
13	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_121):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
12	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_120):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_119):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
10	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_118):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
9	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_117):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
8	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_116):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_115):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
6	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_114):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
5	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_113):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
4	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_112):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_111):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
2	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_110):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
1	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_109):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
0	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_108):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.364 NMI Enable (GPI_NMI_EN_north_2)—Offset 1D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_155):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
14	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_154):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
13	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_153):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
12	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_152):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_151):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
10	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_150):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
9	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_149):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
8	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_148):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_147):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
6	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_146):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
5	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_145):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
4	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_144):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_143):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
2	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_142):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
1	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_141):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
0	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_140):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.365 Event Trigger Output Enable (EVOUTEN_0)—Offset 210h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15:0	0h RW	**EV Trigger Output Enable (EVOUTEN):** When a bit is '0', the corresponding output is masked to '0' I.e. no valid pin assigned to. When a bit is set to '1' the corresponding output is enabled and depends on the EVMAP settings.

29.366 Event Trigger Output Enable (EVOUTEN_1)—Offset 214h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15:0	0h RW	**EV Trigger Output Enable (EVOUTEN):** When a bit is '0', the corresponding output is masked to '0' I.e. no valid pin assigned to. When a bit is set to '1' the corresponding output is enabled and depends on the EVMAP settings.

29.367 Event Trigger Mapping (EVMAP_0)—Offset 220h

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RO	**Reserved (RSVD_0):** Reserved
62:60	0h RW	**Event Trigger Mapping to PMU[15] (EVMAPPM_15):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
59	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
58:56	0h RW	**Event Trigger Mapping to PMU[14] (EVMAPPM_14):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
55	0h RO	**Reserved (RSVD_2):** Reserved
54:52	0h RW	**Event Trigger Mapping to PMU[13] (EVMAPPM_13):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
51	0h RO	**Reserved (RSVD_3):** Reserved
50:48	0h RW	**Event Trigger Mapping to PMU[12] (EVMAPPM_12):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
47	0h RO	**Reserved (RSVD_4):** Reserved
46:44	0h RW	**Event Trigger Mapping to PMU[11] (EVMAPPM_11):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
43	0h RO	**Reserved (RSVD_5):** Reserved
42:40	0h RW	**Event Trigger Mapping to PMU[10] (EVMAPPM_10):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
39	0h RO	**Reserved (RSVD_6):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
38:36	0h RW	**Event Trigger Mapping to PMU[9] (EVMAPPM_9):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
35	0h RO	**Reserved (RSVD_7):** Reserved
34:32	0h RW	**Event Trigger Mapping to PMU[8] (EVMAPPM_8):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
31	0h RO	**Reserved (RSVD_8):** Reserved
30:28	0h RW	**Event Trigger Mapping to PMU[7] (EVMAPPM_7):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
27	0h RO	**Reserved (RSVD_9):** Reserved
26:24	0h RW	**Event Trigger Mapping to PMU[6] (EVMAPPM_6):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
23	0h RO	**Reserved (RSVD_10):** Reserved
22:20	0h RW	**Event Trigger Mapping to PMU[5] (EVMAPPM_5):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
19	0h RO	**Reserved (RSVD_11):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
18:16	0h RW	**Event Trigger Mapping to PMU[4] (EVMAPPM_4):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
15	0h RO	**Reserved (RSVD_12):** Reserved
14:12	0h RW	**Event Trigger Mapping to PMU[3] (EVMAPPM_3):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
11	0h RO	**Reserved (RSVD_13):** Reserved
10:8	0h RW	**Event Trigger Mapping to PMU[2] (EVMAPPM_2):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
7	0h RO	**Reserved (RSVD_14):** Reserved
6:4	0h RW	**Event Trigger Mapping to PMU[1] (EVMAPPM_1):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
3	0h RO	**Reserved (RSVD_15):** Reserved
2:0	0h RW	**Event Trigger Mapping to PMU[0] (EVMAPPM_0):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community

29.368 Event Trigger Mapping (EVMAP_1)—Offset 228h

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RO	**Reserved (RSVD_0):** Reserved
62:60	0h RW	**Event Trigger Mapping to PMU[15] (EVMAPPM_15):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
59	0h RO	**Reserved (RSVD_1):** Reserved
58:56	0h RW	**Event Trigger Mapping to PMU[14] (EVMAPPM_14):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
55	0h RO	**Reserved (RSVD_2):** Reserved
54:52	0h RW	**Event Trigger Mapping to PMU[13] (EVMAPPM_13):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
51	0h RO	**Reserved (RSVD_3):** Reserved
50:48	0h RW	**Event Trigger Mapping to PMU[12] (EVMAPPM_12):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
47	0h RO	**Reserved (RSVD_4):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
46:44	0h RW	**Event Trigger Mapping to PMU[11] (EVMAPPM_11):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
43	0h RO	**Reserved (RSVD_5):** Reserved
42:40	0h RW	**Event Trigger Mapping to PMU[10] (EVMAPPM_10):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
39	0h RO	**Reserved (RSVD_6):** Reserved
38:36	0h RW	**Event Trigger Mapping to PMU[9] (EVMAPPM_9):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
35	0h RO	**Reserved (RSVD_7):** Reserved
34:32	0h RW	**Event Trigger Mapping to PMU[8] (EVMAPPM_8):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
31	0h RO	**Reserved (RSVD_8):** Reserved
30:28	0h RW	**Event Trigger Mapping to PMU[7] (EVMAPPM_7):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
27	0h RO	**Reserved (RSVD_9):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
26:24	0h RW	**Event Trigger Mapping to PMU[6] (EVMAPPM_6):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
23	0h RO	**Reserved (RSVD_10):** Reserved
22:20	0h RW	**Event Trigger Mapping to PMU[5] (EVMAPPM_5):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
19	0h RO	**Reserved (RSVD_11):** Reserved
18:16	0h RW	**Event Trigger Mapping to PMU[4] (EVMAPPM_4):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
15	0h RO	**Reserved (RSVD_12):** Reserved
14:12	0h RW	**Event Trigger Mapping to PMU[3] (EVMAPPM_3):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
11	0h RO	**Reserved (RSVD_13):** Reserved
10:8	0h RW	**Event Trigger Mapping to PMU[2] (EVMAPPM_2):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
7	0h RO	**Reserved (RSVD_14):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
6:4	0h RW	**Event Trigger Mapping to PMU[1] (EVMAPPM_1):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
3	0h RO	**Reserved (RSVD_15):** Reserved
2:0	0h RW	**Event Trigger Mapping to PMU[0] (EVMAPPM_0):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community

29.369 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_SVID)—Offset 300h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	3h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.370 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_SVID)—Offset 304h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RO	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.371 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_SVID)—Offset 308h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RO	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.372 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_SVID)—Offset 30Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RO	**strsel B (STRSELB):** Strsel B value

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.373 Family Configuration Register (FAM_CFG_SVID)—Offset 310h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 5E08000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RW	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
26	1h RW	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
21	1h RO	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	1h RW/V	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	0h RO	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RW	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.374 Family Reserved Register DW5 (FAM_RSVD_DW5SVID)—Offset 314h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.375 Family Reserved Register DW6 (FAM_RSVD_DW6SVID)—Offset 318h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.376 Family Reserved Register DW7 (FAM_RSVD_DW7SVID)—Offset 31Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.377 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_SPI_1P8)—Offset 320h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 13300F3h

Bit Range	Default & Access	Field Name (ID): Description
31:16	133h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	F3h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.378 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_SPI_1P8)—Offset 324h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000202h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	2h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	2h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.379 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_SPI_1P8)—Offset 328h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RO	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.380 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_SPI_1P8)—Offset 32Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RO	**strsel B (STRSELB):** Strsel B value
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.381 Family Configuration Register (FAM_CFG_SPI_1P8)—Offset 330h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1C02000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RO	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
26	0h RO	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	2h RW	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8

Bit Range	Default & Access	Field Name (ID): Description
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RO	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.382 Family Reserved Register DW5 (FAM_RSVD_DW5SPI_1P8)—Offset 334h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.383 Family Reserved Register DW6 (FAM_RSVD_DW6SPI_1P8)—Offset 338h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.384 Family Reserved Register DW7 (FAM_RSVD_DW7SPI_1P8)—Offset 33Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.385 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_PMU_3P3_A)—Offset 340h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.386 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_PMU_3P3_A)—Offset 344h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.387 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_PMU_3P3_A)—Offset 348h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.388 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_PMU_3P3_A)—Offset 34Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RW	**strsel B (STRSELB):** Strsel B value
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.389 Family Configuration Register (FAM_CFG_PMU_3P3_A)—Offset 350h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 7E00000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RW	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
26	1h RW	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	3h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
21	1h RW/V	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RW	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	0h RW	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RW	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.390 Family Reserved Register DW5 (FAM_RSVD_DW5PMU_3P3_A)—Offset 354h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.391 Family Reserved Register DW6 (FAM_RSVD_DW6PMU_3P3_A)—Offset 358h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.392 Family Reserved Register DW7 (FAM_RSVD_DW7PMU_3P3_A)—Offset 35Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.393 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_PMU_3P3_B)—Offset 360h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.394 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_PMU_3P3_B)—Offset 364h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.395 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_PMU_3P3_B)—Offset 368h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.396 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_PMU_3P3_B)—Offset 36Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RW	**strsel B (STRSELB):** Strsel B value

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.397 Family Configuration Register (FAM_CFG_PMU_3P3_B)—Offset 370h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 7E00000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RW	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
26	1h RW	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	3h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
21	1h RW/V	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RW	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	0h RW	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RW	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.398 Family Reserved Register DW5 (FAM_RSVD_DW5PMU_3P3_B)—Offset 374h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.399 Family Reserved Register DW6 (FAM_RSVD_DW6PMU_3P3_B)—Offset 378h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.400 Family Reserved Register DW7 (FAM_RSVD_DW7PMU_3P3_B)—Offset 37Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.401 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_I2C_3P3_A)—Offset 380h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.402 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_I2C_3P3_A)—Offset 384h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.403 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_I2C_3P3_A)—Offset 388h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.404 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_I2C_3P3_A)—Offset 38Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RW	**strsel B (STRSELB):** Strsel B value
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.405 Family Configuration Register (FAM_CFG_I2C_3P3_A)—Offset 390h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 5E00007h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RO	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
26	1h RW	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
21	1h RW	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RW	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	0h RW	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8

Bit Range	Default & Access	Field Name (ID): Description
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	7h RW	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.406 Family Reserved Register DW5 (FAM_RSVD_DW5I2C_3P3_A)—Offset 394h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.407 Family Reserved Register DW6 (FAM_RSVD_DW6I2C_3P3_A)—Offset 398h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.408 Family Reserved Register DW7 (FAM_RSVD_DW7I2C_3P3_A)—Offset 39Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.409 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_I2C_3P3_B)—Offset 3A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.410 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_I2C_3P3_B)—Offset 3A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.411 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_I2C_3P3_B)—Offset 3A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.412 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_I2C_3P3_B)—Offset 3ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RW	**strsel B (STRSELB):** Strsel B value
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.413 Family Configuration Register (FAM_CFG_I2C_3P3_B)—Offset 3B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 5E00007h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RO	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
26	1h RW	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
21	1h RW	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RW	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	0h RW	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	7h RW	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.414 Family Reserved Register DW5 (FAM_RSVD_DW5I2C_3P3_B)—Offset 3B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.415 Family Reserved Register DW6 (FAM_RSVD_DW6I2C_3P3_B)—Offset 3B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.416 Family Reserved Register DW7 (FAM_RSVD_DW7I2C_3P3_B)—Offset 3BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.417 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_PCIE_3P3)—Offset 3C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.418 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_PCIE_3P3)—Offset 3C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.419 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_PCIE_3P3)—Offset 3C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.420 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_PCIE_3P3)—Offset 3CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RW	**strsel B (STRSELB):** Strsel B value

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.421 Family Configuration Register (FAM_CFG_PCIE_3P3)—Offset 3D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 5E00000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RW	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
26	1h RW	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
21	1h RW	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RW	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	0h RW	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RW	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.422 Family Reserved Register DW5 (FAM_RSVD_DW5PCIE_3P3)—Offset 3D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.423 Family Reserved Register DW6 (FAM_RSVD_DW6PCIE_3P3)—Offset 3D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.424 Family Reserved Register DW7 (FAM_RSVD_DW7PCIE_3P3)—Offset 3DCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.425 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_DISP_3P3_A)—Offset 3E0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.426 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_DISP_3P3_A)—Offset 3E4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.427 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_DISP_3P3_A)—Offset 3E8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.428 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_DISP_3P3_A)—Offset 3ECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RW	**strsel B (STRSELB):** Strsel B value
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.429 Family Configuration Register (FAM_CFG_DISP_3P3_A)—Offset 3F0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 5E00000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RW	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
26	1h RW	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
21	1h RW	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RW	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	0h RW	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8

Bit Range	Default & Access	Field Name (ID): Description
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RW	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.430 Family Reserved Register DW5 (FAM_RSVD_DW5DISP_3P3_A)—Offset 3F4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.431 Family Reserved Register DW6 (FAM_RSVD_DW6DISP_3P3_A)—Offset 3F8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.432 Family Reserved Register DW7 (FAM_RSVD_DW7DISP_3P3_A)—Offset 3FCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.433 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_DISP_3P3_B)—Offset 400h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.434 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_DISP_3P3_B)—Offset 404h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.435 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_DISP_3P3_B)—Offset 408h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.436 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_DISP_3P3_B)—Offset 40Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RW	**strsel B (STRSELB):** Strsel B value
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.437 Family Configuration Register (FAM_CFG_DISP_3P3_B)—Offset 410h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 5E00000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RW	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
26	1h RW	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
21	1h RW	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RW	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	0h RW	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RW	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.438 Family Reserved Register DW5 (FAM_RSVD_DW5DISP_3P3_B)—Offset 414h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.439 Family Reserved Register DW6 (FAM_RSVD_DW6DISP_3P3_B)—Offset 418h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.440 Family Reserved Register DW7 (FAM_RSVD_DW7DISP_3P3_B)—Offset 41Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.441 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_GPHS3_3P3_A)—Offset 420h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.442 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_GPHS3_3P3_A)—Offset 424h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.443 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_GPHS3_3P3_A)—Offset 428h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.444 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_GPHS3_3P3_A)—Offset 42Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 30000h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	3h RW	**strsel B (STRSELB):** Strsel B value

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.445 Family Configuration Register (FAM_CFG_GPHS3_3P3_A)—Offset 430h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 5E03000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RW	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
26	1h RW	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
21	1h RW	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RW	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	3h RW	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RW	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.446 Family Reserved Register DW5 (FAM_RSVD_DW5GPHS3_3P3_A)—Offset 434h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.447 Family Reserved Register DW6 (FAM_RSVD_DW6GPHS3_3P3_A)—Offset 438h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.448 Family Reserved Register DW7 (FAM_RSVD_DW7GPHS3_3P3_A)—Offset 43Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.449 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_GPHS3_3P3_B)—Offset 440h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.450 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_GPHS3_3P3_B)—Offset 444h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.451 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_GPHS3_3P3_B)—Offset 448h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.452 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_GPHS3_3P3_B)—Offset 44Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 30000h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	3h RW	**strsel B (STRSELB):** Strsel B value
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.453 Family Configuration Register (FAM_CFG_GPHS3_3P3_B)—Offset 450h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 5E03000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RW	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
26	1h RW	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
21	1h RW	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RW	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	3h RW	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8

Bit Range	Default & Access	Field Name (ID): Description
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RW	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.454 Family Reserved Register DW5 (FAM_RSVD_DW5GPHS3_3P3_B)—Offset 454h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.455 Family Reserved Register DW6 (FAM_RSVD_DW6GPHS3_3P3_B)—Offset 458h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.456 Family Reserved Register DW7 (FAM_RSVD_DW7GPHS3_3P3_B)—Offset 45Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.457 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_LPC_3P3)—Offset 460h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.458 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_LPC_3P3)—Offset 464h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0001818h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	18h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	18h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.459 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_LPC_3P3)—Offset 468h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.460 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_LPC_3P3)—Offset 46Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1818h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RW	**strsel B (STRSELB):** Strsel B value
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	18h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	18h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.461 Family Configuration Register (FAM_CFG_LPC_3P3)—Offset 470h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 7E00000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RW	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
26	1h RW	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	3h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
21	1h RW/V	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RW	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	0h RW	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RW	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.462 Family Reserved Register DW5 (FAM_RSVD_DW5LPC_3P3)—Offset 474h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.463 Family Reserved Register DW6 (FAM_RSVD_DW6LPC_3P3)—Offset 478h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.464 Family Reserved Register DW7 (FAM_RSVD_DW7LPC_3P3)—Offset 47Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.465 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_76)—Offset 600h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).

Bit Range	Default & Access	Field Name (ID): Description
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.466 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_76)—Offset 604h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.467 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_76)—Offset 608h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.468 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_76)—Offset 60Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.469 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_77)—Offset 610h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.470 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_77)—Offset 614h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.471 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_77)—Offset 618h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.472 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_77)—Offset 61Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.473 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_78)—Offset 620h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.474 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_78)—Offset 624h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.475 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_78)—Offset 628h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.476 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_78)—Offset 62Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.477 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_79)—Offset 630h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.478 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_79)—Offset 634h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1032h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	32h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.479 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_79)—Offset 638h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.480 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_79)—Offset 63Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.481 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_80)—Offset 640h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000300h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.482 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_80)—Offset 644h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1033h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	33h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.483 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_80)—Offset 648h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.484 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_80)—Offset 64Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.485 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_81)—Offset 650h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000300h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.486 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_81)—Offset 654h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1034h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	34h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.487 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_81)—Offset 658h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 800h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	1h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.488 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_81)—Offset 65Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.489 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_82)—Offset 660h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.490 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_82)—Offset 664h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1035h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	35h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.491 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_82)—Offset 668h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.492 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_82)—Offset 66Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.493 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_83)—Offset 670h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.494 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_83)—Offset 674h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1036h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	36h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.495 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_83)—Offset 678h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 800h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	1h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.496 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_83)—Offset 67Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.497 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_84)—Offset 680h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.498 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_84)—Offset 684h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1037h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	37h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.499 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_84)—Offset 688h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 800h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	1h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.500 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_84)—Offset 68Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.501 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_85)—Offset 690h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000300h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.502 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_85)—Offset 694h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1038h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	38h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.503 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_85)—Offset 698h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.504 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_85)—Offset 69Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.505 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_86)—Offset 6A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000300h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.506 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_86)—Offset 6A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1039h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	39h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.507 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_86)—Offset 6A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.508 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_86)—Offset 6ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.509 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_87)—Offset 6B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000300h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.510 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_87)—Offset 6B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 103Ah

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	3Ah RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.511 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_87)—Offset 6B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.512 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_87)—Offset 6BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.513 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_88)—Offset 6C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.514 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_88)—Offset 6C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 103Bh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	3Bh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.515 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_88)—Offset 6C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.516 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_88)—Offset 6CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.517 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_89)—Offset 6D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.518 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_89)—Offset 6D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 103Ch

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	3Ch RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.519 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_89)—Offset 6D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.520 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_89)—Offset 6DCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.521 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_90)—Offset 6E0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.522 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_90)—Offset 6E4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3C3Dh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Fh RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	3Dh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.523 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_90)—Offset 6E8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.524 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_90)—Offset 6ECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.525 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_91)—Offset 6F0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.526 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_91)—Offset 6F4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3C3Eh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Fh RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	3Eh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.527 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_91)—Offset 6F8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.528 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_91)—Offset 6FCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.529 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_92)—Offset 700h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.530 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_92)—Offset 704h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3C3Fh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Fh RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	3Fh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.531 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_92)—Offset 708h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.532 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_92)—Offset 70Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.533 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_93)—Offset 710h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.534 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_93)—Offset 714h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3C40h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Fh RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	40h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.535 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_93)—Offset 718h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.536 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_93)—Offset 71Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.537 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_94)—Offset 720h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.538 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_94)—Offset 724h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3C41h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Fh RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	41h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.539 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_94)—Offset 728h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.540 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_94)—Offset 72Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.541 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_95)—Offset 730h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.542 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_95)—Offset 734h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3C42h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Fh RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	42h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.543 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_95)—Offset 738h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.544 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_95)—Offset 73Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.545 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_96)—Offset 740h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.546 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_96)—Offset 744h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3C43h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Fh RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	43h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.547 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_96)—Offset 748h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.548 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_96)—Offset 74Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.549 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_97)—Offset 750h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.550 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_97)—Offset 754h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3C00h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Fh RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.551 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_97)—Offset 758h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.552 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_97)—Offset 75Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.553 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_98)—Offset 760h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.554 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_98)—Offset 764h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.555 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_98)—Offset 768h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.556 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_98)—Offset 76Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.557 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_99)—Offset 770h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.558 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_99)—Offset 774h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.559 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_99)—Offset 778h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.560 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_99)—Offset 77Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.561 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_100)—Offset 780h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.562 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_100)—Offset 784h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.563 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_100)—Offset 788h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.564 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_100)—Offset 78Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.565 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_101)—Offset 790h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.566 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_101)—Offset 794h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.567 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_101)—Offset 798h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.568 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_101)—Offset 79Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.569 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_102)—Offset 7A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.570 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_102)—Offset 7A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.571 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_102)—Offset 7A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.572 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_102)—Offset 7ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.573 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_103)—Offset 7B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.574 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_103)—Offset 7B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.575 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_103)—Offset 7B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.576 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_103)—Offset 7BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.577 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_104)—Offset 7C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.578 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_104)—Offset 7C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.579 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_104)—Offset 7C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.580 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_104)—Offset 7CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.581 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_105)—Offset 7D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RO	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.582 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_105)—Offset 7D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1044h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	44h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.583 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_105)—Offset 7D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.584 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_105)—Offset 7DCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.585 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_106)—Offset 7E0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.586 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_106)—Offset 7E4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.587 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_106)—Offset 7E8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.588 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_106)—Offset 7ECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.589 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_107)—Offset 7F0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.590 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_107)—Offset 7F4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.591 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_107)—Offset 7F8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.592 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_107)—Offset 7FCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.593 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_108)—Offset 800h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.594 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_108)—Offset 804h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.595 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_108)—Offset 808h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.596 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_108)—Offset 80Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.597 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_109)—Offset 810h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.598 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_109)—Offset 814h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.599 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_109)—Offset 818h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.600 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_109)—Offset 81Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.601 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_110)—Offset 820h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.602 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_110)—Offset 824h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3045h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	45h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.603 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_110)—Offset 828h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.604 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_110)—Offset 82Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.605 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_111)—Offset 830h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.606 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_111)—Offset 834h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3046h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	46h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.607 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_111)—Offset 838h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.608 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_111)—Offset 83Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.609 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_112)—Offset 840h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.610 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_112)—Offset 844h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3047h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	47h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.611 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_112)—Offset 848h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.612 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_112)—Offset 84Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.613 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_113)—Offset 850h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.614 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_113)—Offset 854h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3048h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	48h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.615 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_113)—Offset 858h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.616 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_113)—Offset 85Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.617 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_114)—Offset 860h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.618 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_114)—Offset 864h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3049h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	49h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.619 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_114)—Offset 868h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.620 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_114)—Offset 86Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.621 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_115)—Offset 870h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.622 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_115)—Offset 874h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 304Ah

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	4Ah RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.623 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_115)—Offset 878h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.624 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_115)—Offset 87Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.625 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_116)—Offset 880h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RW	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.626 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_116)—Offset 884h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 304Bh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	4Bh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.627 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_116)—Offset 888h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.628 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_116)—Offset 88Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.629 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_117)—Offset 890h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RW	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.630 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_117)—Offset 894h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 304Ch

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	4Ch RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.631 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_117)—Offset 898h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.632 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_117)—Offset 89Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.633 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_118)—Offset 8A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RW	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.634 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_118)—Offset 8A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 304Dh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	4Dh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.635 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_118)—Offset 8A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.636 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_118)—Offset 8ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.637 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_119)—Offset 8B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RW	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.638 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_119)—Offset 8B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 304Eh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	4Eh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.639 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_119)—Offset 8B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.640 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_119)—Offset 8BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.641 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_120)—Offset 8C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RW	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.642 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_120)—Offset 8C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 304Fh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	4Fh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.643 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_120)—Offset 8C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.644 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_120)—Offset 8CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.645 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_121)—Offset 8D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RW	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.646 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_121)—Offset 8D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3050h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	50h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.647 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_121)—Offset 8D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.648 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_121)—Offset 8DCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.649 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_122)—Offset 8E0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RW	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.650 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_122)—Offset 8E4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3051h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	51h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.651 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_122)—Offset 8E8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.652 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_122)—Offset 8ECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.653 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_123)—Offset 8F0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RW	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.654 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_123)—Offset 8F4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3052h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	52h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.655 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_123)—Offset 8F8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.656 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_123)—Offset 8FCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.657 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_124)—Offset 900h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.658 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_124)—Offset 904h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3053h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	53h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.659 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_124)—Offset 908h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.660 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_124)—Offset 90Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.661 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_125)—Offset 910h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.662 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_125)—Offset 914h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3054h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	54h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.663 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_125)—Offset 918h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.664 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_125)—Offset 91Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.665 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_126)—Offset 920h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.666 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_126)—Offset 924h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3055h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	55h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.667 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_126)—Offset 928h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.668 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_126)—Offset 92Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.669 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_127)—Offset 930h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.670 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_127)—Offset 934h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3056h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	56h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.671 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_127)—Offset 938h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.672 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_127)—Offset 93Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.673 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_128)—Offset 940h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.674 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_128)—Offset 944h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1057h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	57h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.675 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_128)—Offset 948h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.676 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_128)—Offset 94Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.677 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_129)—Offset 950h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.678 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_129)—Offset 954h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1058h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	58h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.679 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_129)—Offset 958h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.680 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_129)—Offset 95Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.681 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_130)—Offset 960h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.682 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_130)—Offset 964h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1059h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	59h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.683 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_130)—Offset 968h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.684 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_130)—Offset 96Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.685 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_131)—Offset 970h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.686 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_131)—Offset 974h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 105Ah

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	5Ah RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.687 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_131)—Offset 978h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.688 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_131)—Offset 97Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.689 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_132)—Offset 980h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.690 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_132)—Offset 984h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 105Bh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	5Bh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.691 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_132)—Offset 988h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.692 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_132)—Offset 98Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.693 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_133)—Offset 990h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.694 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_133)—Offset 994h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 105Ch

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	5Ch RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.695 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_133)—Offset 998h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.696 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_133)—Offset 99Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.697 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_134)—Offset 9A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:13	0h RO	**Reserved (RSVD_1):** Reserved
12:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.698 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_134)—Offset 9A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 105Dh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	5Dh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.699 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_134)—Offset 9A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.700 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_134)—Offset 9ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.701 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_135)—Offset 9B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:13	0h RO	**Reserved (RSVD_1):** Reserved
12:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.702 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_135)—Offset 9B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 105Eh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	5Eh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.703 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_135)—Offset 9B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.704 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_135)—Offset 9BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.705 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_136)—Offset 9C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:13	0h RO	**Reserved (RSVD_1):** Reserved
12:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.706 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_136)—Offset 9C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 105Fh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	5Fh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.707 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_136)—Offset 9C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.708 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_136)—Offset 9CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.709 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_137)—Offset 9D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:13	0h RO	**Reserved (RSVD_1):** Reserved
12:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.710 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_137)—Offset 9D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1060h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	60h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.711 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_137)—Offset 9D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.712 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_137)—Offset 9DCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.713 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_138)—Offset 9E0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:13	0h RO	**Reserved (RSVD_1):** Reserved
12:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.714 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_138)—Offset 9E4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1061h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	61h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.715 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_138)—Offset 9E8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.716 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_138)—Offset 9ECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.717 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_139)—Offset 9F0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:13	0h RO	**Reserved (RSVD_1):** Reserved
12:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.718 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_139)—Offset 9F4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1062h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	62h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.719 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_139)—Offset 9F8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.720 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_139)—Offset 9FCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.721 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_140)—Offset A00h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:13	0h RO	**Reserved (RSVD_1):** Reserved
12:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.722 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_140)—Offset A04h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3063h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	63h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.723 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_140)—Offset A08h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.724 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_140)—Offset A0Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.725 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_141)—Offset A10h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:13	0h RO	**Reserved (RSVD_1):** Reserved
12:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.726 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_141)—Offset A14h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3064h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	64h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.727 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_141)—Offset A18h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.728 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_141)—Offset A1Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.729 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_142)—Offset A20h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:13	0h RO	**Reserved (RSVD_1):** Reserved
12:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.730 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_142)—Offset A24h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3065h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	65h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.731 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_142)—Offset A28h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.732 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_142)—Offset A2Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.733 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_143)—Offset A30h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:13	0h RO	**Reserved (RSVD_1):** Reserved
12:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.734 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_143)—Offset A34h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3066h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	66h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.735 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_143)—Offset A38h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.736 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_143)—Offset A3Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.737 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_144)—Offset A40h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:13	0h RO	**Reserved (RSVD_1):** Reserved
12:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.738 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_144)—Offset A44h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3067h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	67h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.739 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_144)—Offset A48h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.740 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_144)—Offset A4Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.741 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_145)—Offset A50h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:13	0h RO	**Reserved (RSVD_1):** Reserved
12:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.742 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_145)—Offset A54h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3068h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	68h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.743 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_145)—Offset A58h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.744 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_145)—Offset A5Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.745 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_146)—Offset A60h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:13	0h RO	**Reserved (RSVD_1):** Reserved
12:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.746 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_146)—Offset A64h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3069h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	69h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.747 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_146)—Offset A68h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.748 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_146)—Offset A6Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.749 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_147)—Offset A70h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.750 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_147)—Offset A74h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 306Ah

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	6Ah RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.751 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_147)—Offset A78h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.752 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_147)—Offset A7Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.753 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_148)—Offset A80h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.754 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_148)—Offset A84h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 6Bh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	6Bh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.755 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_148)—Offset A88h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.756 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_148)—Offset A8Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.757 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_149)—Offset A90h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.758 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_149)—Offset A94h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 6Ch

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	6Ch RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.759 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_149)—Offset A98h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.760 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_149)—Offset A9Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.761 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_150)—Offset AA0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.762 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_150)—Offset AA4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 306Dh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	6Dh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.763 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_150)—Offset AA8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.764 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_150)—Offset AACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.765 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_151)—Offset AB0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.766 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_151)—Offset AB4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 306Eh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	6Eh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.767 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_151)—Offset AB8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.768 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_151)—Offset ABCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.769 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_152)—Offset AC0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.770 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_152)—Offset AC4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 306Fh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	6Fh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.771 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_152)—Offset AC8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.772 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_152)—Offset ACCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.773 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_153)—Offset AD0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.774 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_153)—Offset AD4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3070h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	70h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.775 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_153)—Offset AD8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.776 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_153)—Offset ADCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.777 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_154)—Offset AE0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.778 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_154)—Offset AE4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3071h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	71h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.779 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_154)—Offset AE8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.780 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_154)—Offset AECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.781 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_155)—Offset AF0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:13	0h RO	**Reserved (RSVD_1):** Reserved
12:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.782 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_155)—Offset AF4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3072h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	72h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.783 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_155)—Offset AF8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.784 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_155)—Offset AFCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.785 Revision ID (REV_ID)—Offset 0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 940000h

Bit Range	Default & Access	Field Name (ID): Description
31:16	94h RO	**Hardware Base Revision ID (HWBASEREVID):** This field is used to indicate the baseline hardware revision. 0091h = RTL release for HAS0.91
15:0	0h RO	**Hardware Sub Revision ID (HWSUBIREVID):** This field is used to indicate the revision of the hardware derived from the baseline.

29.786 Capability List Register (CAP_LIST_0)—Offset 4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD_0):** Reserved
23:16	0h RO	**Capability Identification (CAPID):** A unique identification for the current capability. A value of 0 is reserved, and must not be used by any capability. Note:This field is always 0 for the first Capability List register.
15:0	0h RO	**Next Capability List Pointer (NXTCAPLPTR):** Specify the DW-aligned Pointer/ Address to the next item in this capabilities list and must be 0 if there is no capability at all or this is the last capability in the list.

29.787 Family Base Address (FAMBAR)—Offset 8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 300h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15:0	300h RO	**Family Base Address (FAMBAR):** This field provides the starting byte-align address of Family0 register sets within a Community. It is meant for software to discover from where the very first Family register (i.e. Family0 register) starts to compute the next Families address offsets.

29.788 Pad Base Address (PADBAR)—Offset Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 600h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
15:0	600h RO	**Pad Base Address (PADBAR):** This field provides the starting byte-align address of Pad0 register sets within a Community. It is meant for software to discover from where the very first Pad register (i.e. Pad0 register) starts to compute the next Pad address offsets.

29.789 Miscellaneous Configuration (MISCCFG)—Offset 10h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: E043200h

Bit Range	Default & Access	Field Name (ID): Description
31:24	Eh RW	**GPIO Driver Mode Interrupt Select (GPDMINTSEL):** IRQ globally for all pads (GPI_IS with corresponding GPI_IE enable). 0 = Interrupt Line 0 1 = Interrupt Line 1 ... 255 = Interrupt Line 255
23:20	0h RW	**Reserved (Reserved1):** Reserved
19:16	4h RW	**GPIO Group to GPE_DW2 assignment encoding (GPE0_DW2):** This register assigns a specific GPIO Group to the ACPI GPE0[95:64]. The setting is SoC specific.
15:12	3h RW	**GPIO Group to GPE_DW1 assignment encoding (GPE0_DW1):** This register assigns a specific GPIO Group to the ACPI GPE0[63:32]. The setting is SoC specific.
11:8	2h RW	**GPIO Group to GPE_DW0 assignment encoding (GPE0_DW0):** This register assigns a specific GPIO Group to the ACPI GPE0[31:0]. The setting is SoC specific.
7:6	0h RW	**Reserved (Reserved2):** Reserved
5	0h RW	**GPIO IOSF Sideband Dynamic Partition Clock Gating Enable (GPSIDEDPCGEN):** Specify whether the GPIO Community IOSF sideband clock should take part in partition clock gating 0 = Disable participation in IOSF sideband clock dynamic partition clock gating 1 = Enable participation in IOSF sideband clock dynamic partition clock gating
4	0h RW	**GPIO RCOMP clock Dynamic Local Clock Gating (GPRCOMPCDLCGEN):** Specify whether the GPIO Community should perform local clock gating on RCOMP clock 0 = Disable dynamic RCOMP clock local clock gating 1 = Enable dynamic RCOMP clock local clock gating
3	0h RW	**GPIO RTC clock Dynamic Local Clock Gating (GPRTCDLCGEN):** Specify whether the GPIO Community should perform local clock gating on RTC clock 0 = Disable dynamic RTC clock local clock gating 1 = Enable dynamic RTC clock local clock gating
2	0h RW	**GSX Static Local Clock Gating (GSXSLCGEN):** Specify whether the GSX controller should be statically clock gated for power saving if it is not enabled (even though the capability is available). 0 = Disable static local clock gating 1 = Enable static local clock gating
1	0h RW	**GPIO Dynamic Partition Clock Gating Enable (GPDPCGEN):** Specify whether the GPIO Community should take part in partition clock gating 0 = Disable participation in dynamic partition clock gating 1 = Enable participation in dynamic partition clock gating
0	0h RW	**GPIO Dynamic Local Clock Gating Enable (GPDLCGEN):** Specify whether the GPIO Community should perform local clock gating 0 = Disable dynamic local clock gating 1 = Enable dynamic local clock gating

29.790 Miscellaneous Secured Configuration (MISCSECCFG)—Offset 14h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD_0):** Reserved
7:4	0h RW	**Reserved (Reserved1):** Reserved
3	0h RW/1C/V	**Soft strap pull status bit (SSTRAPPULLSTAT):** Indicate the status of the last soft strap pull operation. If the implementation guarantee the success of the operation, then this bit can be hard-wired to 0. 0 = No error 1 = Error FW writes 1 to this bit to clear. This bit is set to 1 when soft strap pull request has been completed with completion without data (for both successful and unsuccessful completion status) or completion with data but with unsuccessful completion status. This bit will also set to 1 if there is any failure on the security and error checking in the fuse puller such as invalid SAI and etc. Fuse puller will assert puller_error in this case and this will set this bit to 1.
2	0h RW/1C/V	**Soft strap re-pull done bit (SSTRAPPULLDONE):** Indication if the soft strap pull has finished the operation 0 = operation is not done. Either not started or in progress 1 = operation is done/finish. FW writes 1 to this bit to clear.
1	0h RW	**Soft strap re-pull request bit (SSTRAPREPULLREQ):** Specify if fuse puller need to re-pull the soft strap. The SoC/PMC/FW can set this bit to request the community fuse puller to re-pull the soft strap. 0 to 1 transition = soft strap re-pull request trigger. A soft strap re-pull request will not trigger IP_READY message sent as the IP_READY message shall be sent during the initial soft strap pull by HW as part of the reset sequences. The SSTRAPREPULLREQ, SSTRAPPULLDONE and SSTRAPPULLSTAT are only meaningful for community with soft strap pull capability.
0	0h RW	**Event Trigger IOSF-SB Message Initiation Disable (SBTRIGDIS):** This bit disables the following list of IOSF-SB Message initiation. It does not disable the event trigger. When this bit is cleared to '0', any pending message due to event trigger while this bit is '1' shall be sent on IOSF-SB. Event list: 'Virtual Wire' message IRQ GPE/SCI NMI SMI

29.791 Reserved (RSVD0[0])—Offset 18h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.792 Reserved (RSVD0[1])—Offset 1Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.793 Pad Ownership (PAD_OWN_northwest_0)—Offset 20h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_7):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_6):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_5):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_4):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_3):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_2):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_1):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_0):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.794 Pad Ownership (PAD_OWN_northwest_1)—Offset 24h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_15):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_14):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_13):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_12):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_11):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_10):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_9):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_8):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.795 Pad Ownership (PAD_OWN_northwest_2)—Offset 28h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_23):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_22):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_21):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_20):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_19):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_18):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_17):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_16):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.796 Pad Ownership (PAD_OWN_northwest_3)—Offset 2Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_31):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_30):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_29):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_28):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_27):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_26):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_25):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_24):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.797 Pad Ownership (PAD_OWN_northwest_4)—Offset 30h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_39):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_38):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_37):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_36):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_35):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_34):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_33):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_32):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.798 Pad Ownership (PAD_OWN_northwest_5)—Offset 34h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_47):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_46):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_45):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_44):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_43):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_42):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_41):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_40):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.799 Pad Ownership (PAD_OWN_northwest_6)—Offset 38h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_55):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_54):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_53):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_52):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_51):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_50):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_49):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_48):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.800 Pad Ownership (PAD_OWN_northwest_7)—Offset 3Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_63):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_62):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_61):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_60):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_59):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_58):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_57):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_56):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.801 Pad Ownership (PAD_OWN_northwest_8)—Offset 40h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_71):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_70):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_69):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_68):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_67):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_66):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_65):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_64):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.802 Pad Ownership (PAD_OWN_northwest_9)—Offset 44h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_214):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_213):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_212):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_211):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_75):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_74):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_73):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_72):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.803 Pad Ownership (PAD_OWN_northwest_10)—Offset 48h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 10h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:28	0h RW	**Pad Ownership (PAD_OWN_vGPIO_7):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
25:24	0h RW	**Pad Ownership (PAD_OWN_vGPIO_6):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved
21:20	0h RW	**Pad Ownership (PAD_OWN_vGPIO_5):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
17:16	0h RW	**Pad Ownership (PAD_OWN_vGPIO_4):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved
13:12	0h RW	**Pad Ownership (PAD_OWN_vGPIO_3):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW	**Pad Ownership (PAD_OWN_vGPIO_2):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved
5:4	1h RW	**Pad Ownership (PAD_OWN_vGPIO_1):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Pad Ownership (PAD_OWN_vGPIO_0):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.804 Pad Ownership (PAD_OWN_northwest_11)—Offset 4Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
29:28	0h RW	**Pad Ownership (PAD_OWN_vGPIO_15):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved
25:24	0h RW	**Pad Ownership (PAD_OWN_vGPIO_14):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
21:20	0h RW	**Pad Ownership (PAD_OWN_vGPIO_13):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved
17:16	0h RW	**Pad Ownership (PAD_OWN_vGPIO_12):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:12	0h RW	**Pad Ownership (PAD_OWN_vGPIO_11):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved
9:8	0h RW	**Pad Ownership (PAD_OWN_vGPIO_10):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5:4	0h RW	**Pad Ownership (PAD_OWN_vGPIO_9):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved
1:0	0h RW	**Pad Ownership (PAD_OWN_vGPIO_8):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.805 Pad Ownership (PAD_OWN_northwest_12)—Offset 50h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:28	0h RW	**Pad Ownership (PAD_OWN_vGPIO_23):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
25:24	0h RW	**Pad Ownership (PAD_OWN_vGPIO_22):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved
21:20	0h RW	**Pad Ownership (PAD_OWN_vGPIO_21):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
17:16	0h RW	**Pad Ownership (PAD_OWN_vGPIO_20):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved
13:12	0h RW	**Pad Ownership (PAD_OWN_vGPIO_19):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW	**Pad Ownership (PAD_OWN_vGPIO_18):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved
5:4	0h RW	**Pad Ownership (PAD_OWN_vGPIO_17):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Pad Ownership (PAD_OWN_vGPIO_16):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.806 Pad Ownership (PAD_OWN_northwest_13)—Offset 54h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
25:24	0h RW	**Pad Ownership (PAD_OWN_vGPIO_30):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_1):** Reserved
21:20	0h RW	**Pad Ownership (PAD_OWN_vGPIO_29):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
17:16	0h RW	**Pad Ownership (PAD_OWN_vGPIO_28):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_3):** Reserved
13:12	0h RW	**Pad Ownership (PAD_OWN_vGPIO_27):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_4):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW	**Pad Ownership (PAD_OWN_vGPIO_26):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_5):** Reserved
5:4	0h RW	**Pad Ownership (PAD_OWN_vGPIO_25):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_6):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Pad Ownership (PAD_OWN_vGPIO_24):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.807 Reserved (RSVD1[0])—Offset 58h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.808 Reserved (RSVD1[1])—Offset 5Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.809 GPI Virtual Wire Message Enable (GPI_VWE_northwest_0)—Offset 60h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_31):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
30	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_30):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
29	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_29):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
28	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_28):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
27	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_27):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
26	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_26):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
25	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_25):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
24	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_24):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
23	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_23):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
22	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_22):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
21	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_21):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
20	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_20):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
19	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_19):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
18	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_18):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
17	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_17):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
16	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_16):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
15	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_15):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
14	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_14):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
13	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_13):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_12):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_11):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_10):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
9	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_9):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_8):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_7):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_6):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
5	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_5):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_4):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_3):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_2):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_1):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_0):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.810 GPI Virtual Wire Message Enable (GPI_VWE_northwest_1)—Offset 64h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_63):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
30	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_62):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
29	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_61):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
28	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_60):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_59):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
26	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_58):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
25	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_57):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
24	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_56):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
23	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_55):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
22	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_54):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
21	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_53):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
20	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_52):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
19	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_51):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
18	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_50):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
17	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_49):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
16	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_48):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_47):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
14	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_46):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_45):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_44):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
11	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_43):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_42):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
9	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_41):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_40):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_39):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_38):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
5	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_37):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_36):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_35):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_34):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_33):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_32):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.811 GPI Virtual Wire Message Enable (GPI_VWE_northwest_2)—Offset 68h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_15):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
30	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_14):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
29	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_13):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
28	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_12):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
27	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_11):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
26	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_10):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
25	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_9):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
24	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_8):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
23	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_7):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
22	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_6):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
21	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_5):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_4):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
19	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_3):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
18	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_2):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
17	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_1):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
16	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_0):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
15	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_214):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
14	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_213):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_212):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
12	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_211):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_75):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_74):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
9	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_73):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
8	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_72):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_71):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_70):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
5	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_69):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
4	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_68):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_67):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_66):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_65):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_64):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.812 GPI Virtual Wire Message Enable (GPI_VWE_northwest_3)—Offset 6Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD_0):** Reserved
14	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_30):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
13	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_29):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_28):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_27):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_26):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
9	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_25):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_24):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_23):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_22):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
5	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_21):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_20):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_19):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_18):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_17):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_vGPIO_16):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.813 Reserved (RSVD2[0])—Offset 70h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.814 Reserved (RSVD2[1])—Offset 74h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.815 Reserved (RSVD2[2])—Offset 78h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.816 Reserved (RSVD2[3])—Offset 7Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.817 Pad Configuration Lock (PADCFGLOCK_northwest_0)—Offset 80h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_31):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
30	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_30):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
29	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_29):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
28	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_28):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_27):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
26	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_26):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_25):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
24	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_24):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_23):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
22	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_22):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_21):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
20	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_20):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_19):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
18	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_18):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_17):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
16	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_16):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_15):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
14	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_14):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_13):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
12	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_12):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_11):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
10	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_10):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_9):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
8	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_8):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_7):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
6	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_6):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_5):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
4	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_4):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_3):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
2	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_2):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_1):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
0	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_0):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.818 Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_northwest_0)—Offset 84h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_31):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
30	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_30):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
29	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_29):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
28	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_28):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_27):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
26	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_26):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_25):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
24	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_24):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_23):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
22	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_22):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_21):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
20	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_20):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_19):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
18	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_18):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_17):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
16	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_16):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_15):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
14	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_14):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_13):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
12	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_12):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_11):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
10	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_10):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_9):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
8	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_8):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_7):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
6	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_6):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_5):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
4	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_4):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_3):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
2	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_2):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_1):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
0	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_0):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.819 Pad Configuration Lock (PADCFGLOCK_northwest_1)—Offset 88h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_63):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
30	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_62):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
29	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_61):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
28	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_60):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_59):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
26	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_58):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_57):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
24	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_56):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_55):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
22	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_54):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_53):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
20	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_52):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_51):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
18	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_50):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_49):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
16	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_48):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_47):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
14	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_46):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_45):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
12	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_44):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_43):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
10	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_42):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_41):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
8	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_40):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_39):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
6	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_38):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_37):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
4	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_36):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_35):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
2	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_34):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_33):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
0	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_32):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.820 Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_northwest_1)—Offset 8Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_63):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
30	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_62):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
29	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_61):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
28	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_60):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_59):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
26	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_58):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_57):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
24	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_56):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_55):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
22	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_54):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_53):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
20	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_52):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_51):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
18	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_50):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_49):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
16	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_48):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_47):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
14	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_46):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_45):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
12	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_44):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_43):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
10	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_42):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_41):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
8	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_40):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_39):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
6	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_38):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_37):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
4	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_36):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_35):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
2	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_34):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_33):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
0	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_32):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.821 Pad Configuration Lock (PADCFGLOCK_northwest_2)—Offset 90h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_15):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
30	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_14):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
29	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_13):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
28	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_12):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_11):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
26	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_10):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_9):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
24	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_8):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_7):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
22	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_6):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_5):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
20	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_4):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_3):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
18	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_2):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_1):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
16	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_0):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_214):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
14	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_213):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_212):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
12	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_211):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_75):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
10	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_74):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_73):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
8	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_72):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_71):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
6	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_70):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_69):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
4	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_68):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_67):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
2	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_66):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_65):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
0	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_64):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.822 Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_northwest_2)—Offset 94h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_15):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
30	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_14):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
29	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_13):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
28	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_12):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_11):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
26	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_10):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_9):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
24	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_8):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_7):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
22	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_6):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_5):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
20	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_4):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_3):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
18	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_2):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_1):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
16	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_0):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_214):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
14	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_213):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_212):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
12	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_211):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_75):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
10	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_74):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_73):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
8	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_72):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_71):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
6	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_70):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_69):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
4	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_68):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_67):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
2	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_66):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_65):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
0	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_64):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.823 Pad Configuration Lock (PADCFGLOCK_northwest_3)—Offset 98h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD_0):** Reserved
14	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_30):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
13	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_29):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_28):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
11	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_27):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_26):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
9	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_25):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_24):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
7	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_23):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_22):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
5	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_21):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_20):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
3	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_19):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_18):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
1	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_17):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**Pad Config Lock (PADCFGLOCK_vGPIO_16):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.824 Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_northwest_3)—Offset 9Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_30):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
13	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_29):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_28):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
11	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_27):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_26):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
9	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_25):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_24):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
7	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_23):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_22):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
5	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_21):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_20):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
3	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_19):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_18):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
1	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_17):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_vGPIO_16):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.825 Reserved (RSVD3[0])—Offset A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.826 Reserved (RSVD3[1])—Offset A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.827 Reserved (RSVD3[2])—Offset A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.828 Reserved (RSVD3[3])—Offset ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.829 Host Software Pad Ownership (HOSTSW_OWN_northwest_0)—Offset B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_31):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
30	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_30):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
29	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_29):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
28	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_28):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
27	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_27):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
26	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_26):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_25):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
24	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_24):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
23	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_23):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
22	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_22):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
21	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_21):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_20):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
19	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_19):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
18	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_18):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
17	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_17):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
16	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_16):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_15):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
14	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_14):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_13):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_12):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_11):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_10):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_9):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_8):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_7):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
6	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_6):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_5):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_4):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_3):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_2):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_1):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_0):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.830 Host Software Pad Ownership (HOSTSW_OWN_northwest_1)—Offset B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_63):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
30	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_62):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
29	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_61):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
28	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_60):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
27	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_59):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
26	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_58):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
25	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_57):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
24	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_56):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_55):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
22	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_54):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
21	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_53):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
20	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_52):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
19	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_51):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_50):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
17	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_49):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
16	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_48):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
15	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_47):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
14	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_46):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_45):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_44):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_43):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_42):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_41):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_40):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_39):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
6	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_38):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_37):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_36):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_35):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_34):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_33):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_32):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.831 Host Software Pad Ownership (HOSTSW_OWN_northwest_2)—Offset B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_15):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
30	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_14):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
29	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_13):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
28	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_12):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
27	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_11):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_10):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
25	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_9):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
24	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_8):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
23	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_7):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
22	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_6):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_5):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
20	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_4):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
19	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_3):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
18	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_2):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
17	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_1):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_0):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
15	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_214):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
14	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_213):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_212):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_211):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_75):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_74):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_73):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_72):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_71):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_70):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_69):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_68):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_67):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_66):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_65):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_64):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.832 Host Software Pad Ownership (HOSTSW_OWN_northwest_3)—Offset BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD_0):** Reserved
14	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_30):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_29):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_28):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_27):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_26):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_25):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_24):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_23):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
6	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_22):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_21):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_20):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_19):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_18):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_17):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW	**HostSW_Own (HOSTSW_OWN_vGPIO_16):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.833 Reserved (RSVD4[0])—Offset C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.834 Reserved (RSVD4[1])—Offset C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.835 Reserved (RSVD4[2])—Offset C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.836 Reserved (RSVD4[3])—Offset CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.837 Reserved (RSVD4[4])—Offset D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.838 Reserved (RSVD4[5])—Offset D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.839 Reserved (RSVD4[6])—Offset D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.840 Reserved (RSVD4[7])—Offset DCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.841 Reserved (RSVD4[8])—Offset E0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.842 Reserved (RSVD4[9])—Offset E4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.843 Reserved (RSVD4[10])—Offset E8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.844 Reserved (RSVD4[11])—Offset ECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.845 Reserved (RSVD4[12])—Offset F0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.846 Reserved (RSVD4[13])—Offset F4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.847 Reserved (RSVD4[14])—Offset F8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.848 Reserved (RSVD4[15])—Offset FCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.849 GPI Interrupt Status (GPI_IS_northwest_0)—Offset 100h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_31):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
30	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_30):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
29	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_29):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
28	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_28):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
27	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_27):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
26	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_26):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
25	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_25):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
24	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_24):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
23	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_23):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
22	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_22):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
21	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_21):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_20):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
19	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_19):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
18	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_18):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
17	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_17):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_16):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
15	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_15):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
14	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_14):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_13):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_12):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_11):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_10):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_9):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_8):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_7):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
6	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_6):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_5):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_4):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_3):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_2):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_1):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_0):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.850 GPI Interrupt Status (GPI_IS_northwest_1)—Offset 104h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_63):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
30	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_62):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
29	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_61):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
28	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_60):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
27	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_59):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
26	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_58):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_57):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
24	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_56):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
23	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_55):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
22	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_54):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_53):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
20	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_52):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
19	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_51):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
18	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_50):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_49):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
16	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_48):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
15	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_47):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
14	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_46):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_45):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_44):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_43):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_42):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_41):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_40):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_39):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
6	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_38):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_37):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_36):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_35):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_34):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_33):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_32):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.851 GPI Interrupt Status (GPI_IS_northwest_2)—Offset 108h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_15):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
30	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_14):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
29	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_13):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
28	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_12):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
27	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_11):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_10):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
25	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_9):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
24	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_8):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
23	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_7):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
22	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_6):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
21	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_5):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
20	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_4):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
19	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_3):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_2):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
17	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_1):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
16	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_0):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
15	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_214):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_213):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_212):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_211):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_75):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_74):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_73):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_72):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_71):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_70):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_69):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_68):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_67):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_66):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_65):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_64):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.852 GPI Interrupt Status (GPI_IS_northwest_3)—Offset 10Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_30):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_29):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_28):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_27):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_26):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_25):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_24):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_23):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_22):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_21):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_20):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_19):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_18):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_17):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_vGPIO_16):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.853 GPI Interrupt Enable (GPI_IE_northwest_0)—Offset 110h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_31):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
30	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_30):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
29	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_29):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
28	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_28):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
27	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_27):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
26	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_26):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_25):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
24	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_24):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
23	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_23):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
22	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_22):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
21	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_21):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
20	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_20):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_19):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
18	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_18):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
17	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_17):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
16	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_16):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
15	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_15):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
14	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_14):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_13):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_12):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_11):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_10):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_9):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_8):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_7):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
6	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_6):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_5):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_4):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_3):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_2):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_1):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_0):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.854 GPI Interrupt Enable (GPI_IE_northwest_1)—Offset 114h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_63):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
30	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_62):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
29	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_61):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
28	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_60):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
27	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_59):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
26	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_58):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
25	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_57):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
24	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_56):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_55):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
22	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_54):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
21	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_53):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
20	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_52):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
19	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_51):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
18	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_50):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_49):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
16	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_48):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
15	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_47):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
14	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_46):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_45):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_44):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_43):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_42):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_41):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_40):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_39):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
6	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_38):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_37):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_36):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_35):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_34):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_33):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_32):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.855 GPI Interrupt Enable (GPI_IE_northwest_2)—Offset 118h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_15):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
30	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_14):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
29	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_13):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
28	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_12):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
27	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_11):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_10):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
25	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_9):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
24	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_8):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
23	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_7):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
22	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_6):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
21	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_5):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_4):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
19	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_3):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
18	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_2):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
17	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_1):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
16	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_0):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
15	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_214):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_213):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_212):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_211):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_75):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_74):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_73):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_72):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_71):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
6	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_70):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_69):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_68):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_67):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_66):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_65):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_64):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.856 GPI Interrupt Enable (GPI_IE_northwest_3)—Offset 11Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD_0):** Reserved
14	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_30):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_29):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_28):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_27):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_26):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_25):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_24):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_23):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
6	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_22):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_21):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_20):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_19):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_18):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_17):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW	**GPI Interrupt Enable (GPI_INT_EN_vGPIO_16):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.857 Reserved (RSVD5[0])—Offset 120h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.858 Reserved (RSVD5[1])—Offset 124h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.859 Reserved (RSVD5[2])—Offset 128h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.860 Reserved (RSVD5[3])—Offset 12Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.861 GPI General Purpose Events Status (GPI_GPE_STS_northwest_0)—Offset 130h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_31):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
30	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_30):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
29	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_29):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
28	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_28):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
27	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_27):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_26):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
25	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_25):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
24	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_24):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
23	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_23):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
22	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_22):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
21	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_21):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
20	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_20):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
19	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_19):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_18):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
17	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_17):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
16	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_16):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
15	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_15):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_14):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_13):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_12):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_11):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_10):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
9	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_9):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_8):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_7):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
6	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_6):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
5	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_5):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_4):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_3):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
2	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_2):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_1):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_0):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.862 GPI General Purpose Events Status (GPI_GPE_STS_northwest_1)—Offset 134h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_63):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
30	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_62):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
29	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_61):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
28	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_60):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_59):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
26	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_58):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
25	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_57):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
24	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_56):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_55):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
22	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_54):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
21	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_53):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
20	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_52):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_51):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
18	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_50):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
17	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_49):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
16	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_48):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_47):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
14	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_46):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_45):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_44):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_43):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_42):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
9	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_41):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_40):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_39):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_38):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
5	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_37):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_36):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_35):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_34):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_33):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_32):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.863 GPI General Purpose Events Status (GPI_GPE_STS_northwest_2)—Offset 138h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_15):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
30	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_14):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
29	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_13):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
28	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_12):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
27	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_11):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
26	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_10):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
25	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_9):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
24	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_8):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
23	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_7):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
22	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_6):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
21	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_5):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_4):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
19	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_3):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
18	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_2):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
17	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_1):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
16	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_0):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
15	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_214):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
14	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_213):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_212):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_211):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_75):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_74):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
9	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_73):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
8	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_72):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_71):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_70):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
5	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_69):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_68):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_67):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_66):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_65):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_64):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.864 GPI General Purpose Events Status (GPI_GPE_STS_northwest_3)—Offset 13Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD_0):** Reserved
14	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_30):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
13	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_29):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_28):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_27):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_26):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
9	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_25):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_24):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_23):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_22):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
5	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_21):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_20):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_19):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_18):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_17):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_vGPIO_16):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.865 Reserved (RSVD6[0])—Offset 140h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.866 Reserved (RSVD6[1])—Offset 144h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.867 Reserved (RSVD6[2])—Offset 148h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.868 Reserved (RSVD6[3])—Offset 14Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.869 GPI General Purpose Events Enable (GPI_GPE_EN_northwest_0)—Offset 150h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_31):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
30	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_30):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
29	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_29):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
28	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_28):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
27	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_27):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_26):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
25	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_25):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
24	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_24):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
23	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_23):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
22	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_22):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_21):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
20	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_20):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
19	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_19):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
18	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_18):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
17	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_17):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_16):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
15	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_15):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
14	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_14):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_13):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_12):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_11):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_10):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
9	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_9):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_8):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_7):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
6	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_6):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
5	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_5):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_4):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_3):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_2):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_1):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_0):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.870 GPI General Purpose Events Enable (GPI_GPE_EN_northwest_1)—Offset 154h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_63):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
30	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_62):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
29	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_61):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
28	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_60):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
27	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_59):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
26	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_58):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_57):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
24	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_56):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
23	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_55):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
22	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_54):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
21	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_53):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_52):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
19	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_51):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
18	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_50):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
17	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_49):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
16	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_48):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_47):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
14	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_46):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_45):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_44):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_43):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
10	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_42):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
9	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_41):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_40):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_39):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_38):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_37):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_36):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_35):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_34):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_33):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_32):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.871 GPI General Purpose Events Enable (GPI_GPE_EN_northwest_2)—Offset 158h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_15):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
30	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_14):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
29	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_13):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
28	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_12):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
27	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_11):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
26	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_10):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
25	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_9):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
24	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_8):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
23	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_7):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
22	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_6):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
21	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_5):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
20	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_4):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
19	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_3):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
18	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_2):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
17	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_1):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
16	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_0):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
15	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_214):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_213):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_212):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_211):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_75):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_74):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_73):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_72):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_71):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_70):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
5	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_69):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_68):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_67):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_66):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_65):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_64):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.872 GPI General Purpose Events Enable (GPI_GPE_EN_northwest_3)—Offset 15Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD_0):** Reserved
14	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_30):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_29):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_28):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_27):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
10	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_26):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
9	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_25):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_24):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_23):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_22):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
5	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_21):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_20):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_19):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_18):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_17):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_vGPIO_16):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.873 Reserved (RSVD7[0])—Offset 160h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.874 Reserved (RSVD7[1])—Offset 164h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.875 Reserved (RSVD7[2])—Offset 168h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.876 Reserved (RSVD7[3])—Offset 16Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.877 SMI Status (GPI_SMI_STS_northwest_0)—Offset 170h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_31):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
30	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_30):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
29	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_29):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
28	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_28):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_27):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
26	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_26):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
25	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_25):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
24	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_24):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
23	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_23):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
22	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_22):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_21):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
20	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_20):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
19	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_19):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_18):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
17	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_17):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
16	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_16):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_15):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
14	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_14):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
13	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_13):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_12):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
11	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_11):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
10	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_10):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_9):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
8	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_8):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
7	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_7):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_6):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
5	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_5):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
4	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_4):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_3):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
2	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_2):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
1	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_1):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_0):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

29.878 SMI Status (GPI_SMI_STS_northwest_1)—Offset 174h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_63):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
30	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_62):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
29	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_61):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
28	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_60):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_59):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
26	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_58):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
25	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_57):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
24	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_56):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
23	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_55):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
22	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_54):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_53):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
20	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_52):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
19	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_51):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_50):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
17	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_49):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
16	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_48):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_47):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
14	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_46):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
13	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_45):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_44):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
11	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_43):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
10	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_42):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_41):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
8	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_40):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
7	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_39):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_38):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
5	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_37):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
4	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_36):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_35):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
2	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_34):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
1	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_33):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_32):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

29.879 SMI Status (GPI_SMI_STS_northwest_2)—Offset 178h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_15):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
30	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_14):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
29	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_13):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
28	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_12):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_11):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
26	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_10):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
25	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_9):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
24	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_8):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
23	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_7):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
22	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_6):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_5):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
20	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_4):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
19	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_3):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_2):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
17	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_1):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
16	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_0):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_214):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
14	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_213):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
13	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_212):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_211):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
11	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_75):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
10	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_74):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_73):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
8	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_72):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
7	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_71):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_70):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
5	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_69):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
4	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_68):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_67):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
2	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_66):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
1	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_65):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_64):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

29.880 SMI Status (GPI_SMI_STS_northwest_3)—Offset 17Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD_0):** Reserved
14	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_30):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_29):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
12	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_28):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
11	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_27):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_26):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
9	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_25):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
8	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_24):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_23):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
6	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_22):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
5	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_21):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_20):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
3	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_19):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
2	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_18):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_17):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
0	0h RO	**GPI SMI Status (GPI_SMI_STS_vGPIO_16):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

29.881 Reserved (RSVD8[0])—Offset 180h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.882 Reserved (RSVD8[1])—Offset 184h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.883 Reserved (RSVD8[2])—Offset 188h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.884 Reserved (RSVD8[3])—Offset 18Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.885 SMI Enable (GPI_SMI_EN_northwest_0)—Offset 190h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_31):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
30	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_30):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
29	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_29):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
28	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_28):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_27):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
26	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_26):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
25	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_25):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
24	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_24):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_23):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
22	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_22):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
21	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_21):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
20	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_20):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_19):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
18	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_18):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
17	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_17):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
16	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_16):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_15):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
14	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_14):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
13	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_13):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
12	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_12):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_11):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
10	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_10):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
9	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_9):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
8	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_8):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_7):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
6	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_6):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
5	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_5):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
4	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_4):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_3):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
2	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_2):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
1	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_1):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
0	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_0):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

29.886 SMI Enable (GPI_SMI_EN_northwest_1)—Offset 194h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_63):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
30	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_62):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
29	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_61):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
28	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_60):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_59):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
26	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_58):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
25	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_57):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
24	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_56):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_55):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
22	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_54):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
21	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_53):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
20	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_52):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_51):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
18	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_50):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
17	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_49):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
16	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_48):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_47):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
14	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_46):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
13	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_45):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
12	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_44):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_43):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
10	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_42):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
9	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_41):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
8	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_40):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_39):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
6	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_38):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
5	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_37):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
4	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_36):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_35):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
2	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_34):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
1	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_33):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
0	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_32):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

29.887 SMI Enable (GPI_SMI_EN_northwest_2)—Offset 198h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_15):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
30	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_14):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
29	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_13):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
28	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_12):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_11):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
26	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_10):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
25	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_9):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
24	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_8):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
23	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_7):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
22	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_6):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
21	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_5):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
20	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_4):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
19	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_3):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
18	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_2):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
17	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_1):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
16	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_0):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_214):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
14	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_213):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
13	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_212):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
12	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_211):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
11	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_75):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
10	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_74):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
9	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_73):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
8	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_72):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_71):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
6	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_70):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
5	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_69):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
4	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_68):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_67):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
2	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_66):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
1	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_65):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
0	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_64):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

29.888 SMI Enable (GPI_SMI_EN_northwest_3)—Offset 19Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD_0):** Reserved
14	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_30):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
13	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_29):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
12	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_28):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
11	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_27):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
10	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_26):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
9	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_25):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
8	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_24):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_23):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
6	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_22):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
5	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_21):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
4	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_20):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_19):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
2	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_18):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
1	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_17):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
0	0h RO	**GPI SMI Enable (GPI_SMI_EN_vGPIO_16):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

29.889 Reserved (RSVD9[0])—Offset 1A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.890 Reserved (RSVD9[1])—Offset 1A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.891 Reserved (RSVD9[2])—Offset 1A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.892 Reserved (RSVD9[3])—Offset 1ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.893 NMI Status (GPI_NMI_STS_northwest_0)—Offset 1B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_31):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
30	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_30):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
29	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_29):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
28	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_28):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
27	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_27):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
26	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_26):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
25	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_25):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
24	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_24):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
23	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_23):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
22	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_22):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
21	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_21):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_20):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
19	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_19):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
18	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_18):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
17	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_17):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
16	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_16):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
15	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_15):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
14	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_14):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
13	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_13):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_12):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
11	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_11):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
10	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_10):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
9	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_9):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
8	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_8):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
7	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_7):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
6	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_6):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
5	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_5):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_4):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
3	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_3):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
2	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_2):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
1	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_1):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_0):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.894 NMI Status (GPI_NMI_STS_northwest_1)—Offset 1B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_63):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
30	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_62):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
29	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_61):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
28	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_60):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
27	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_59):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
26	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_58):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_57):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
24	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_56):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
23	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_55):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
22	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_54):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_53):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
20	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_52):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
19	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_51):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
18	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_50):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
17	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_49):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
16	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_48):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
15	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_47):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
14	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_46):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_45):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
12	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_44):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
11	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_43):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
10	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_42):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_41):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
8	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_40):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
7	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_39):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
6	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_38):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_37):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
4	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_36):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
3	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_35):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
2	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_34):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_33):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
0	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_32):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.895 NMI Status (GPI_NMI_STS_northwest_2)—Offset 1B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_15):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
30	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_14):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
29	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_13):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
28	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_12):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
27	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_11):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
26	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_10):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
25	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_9):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
24	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_8):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
23	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_7):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
22	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_6):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
21	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_5):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
20	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_4):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
19	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_3):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_2):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
17	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_1):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
16	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_0):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
15	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_214):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
14	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_213):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
13	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_212):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
12	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_211):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
11	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_75):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_74):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
9	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_73):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
8	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_72):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
7	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_71):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_70):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
5	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_69):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
4	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_68):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
3	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_67):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_66):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
1	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_65):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
0	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_64):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.896 NMI Status (GPI_NMI_STS_northwest_3)—Offset 1BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
14	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_30):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
13	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_29):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
12	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_28):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
11	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_27):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_26):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
9	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_25):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
8	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_24):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
7	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_23):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_22):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
5	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_21):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
4	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_20):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
3	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_19):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_18):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
1	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_17):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
0	0h RO	**GPI NMI Status (GPI_NMI_STS_vGPIO_16):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.897 Reserved (RSVD10[0])—Offset 1C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.898 Reserved (RSVD10[1])—Offset 1C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.899 Reserved (RSVD10[2])—Offset 1C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.900 Reserved (RSVD10[3])—Offset 1CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.901 NMI Enable (GPI_NMI_EN_northwest_0)—Offset 1D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_31):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
30	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_30):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
29	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_29):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
28	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_28):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_27):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
26	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_26):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
25	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_25):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
24	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_24):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_23):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
22	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_22):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
21	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_21):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
20	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_20):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_19):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
18	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_18):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
17	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_17):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
16	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_16):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_15):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
14	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_14):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
13	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_13):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
12	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_12):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_11):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
10	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_10):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
9	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_9):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
8	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_8):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_7):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
6	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_6):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
5	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_5):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
4	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_4):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_3):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
2	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_2):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
1	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_1):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
0	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_0):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.902 NMI Enable (GPI_NMI_EN_northwest_1)—Offset 1D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_63):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
30	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_62):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
29	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_61):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
28	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_60):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_59):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
26	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_58):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
25	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_57):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
24	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_56):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_55):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
22	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_54):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
21	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_53):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
20	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_52):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_51):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
18	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_50):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
17	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_49):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
16	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_48):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_47):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
14	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_46):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
13	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_45):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
12	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_44):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_43):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
10	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_42):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
9	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_41):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
8	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_40):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_39):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
6	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_38):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
5	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_37):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
4	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_36):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_35):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
2	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_34):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
1	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_33):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
0	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_32):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.903 NMI Enable (GPI_NMI_EN_northwest_2)—Offset 1D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_15):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
30	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_14):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
29	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_13):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
28	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_12):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_11):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
26	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_10):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
25	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_9):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
24	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_8):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_7):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
22	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_6):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
21	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_5):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
20	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_4):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_3):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
18	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_2):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
17	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_1):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
16	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_0):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_214):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
14	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_213):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
13	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_212):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
12	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_211):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_75):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
10	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_74):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
9	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_73):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
8	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_72):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_71):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
6	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_70):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
5	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_69):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
4	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_68):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_67):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
2	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_66):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
1	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_65):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
0	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_64):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.904 NMI Enable (GPI_NMI_EN_northwest_3)—Offset 1DCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:15	0h RO	**Reserved (RSVD_0):** Reserved
14	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_30):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
13	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_29):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
12	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_28):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
11	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_27):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
10	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_26):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
9	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_25):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
8	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_24):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
7	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_23):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_22):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
5	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_21):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
4	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_20):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
3	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_19):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
2	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_18):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
1	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_17):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
0	0h RO	**GPI NMI Enable (GPI_NMI_EN_vGPIO_16):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.905 Reserved (RSVD11[0])—Offset 1E0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.906 Reserved (RSVD11[1])—Offset 1E4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.907 Reserved (RSVD11[2])—Offset 1E8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.908 Reserved (RSVD11[3])—Offset 1ECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.909 Reserved (RSVD11[4])—Offset 1F0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.910 Reserved (RSVD11[5])—Offset 1F4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.911 Reserved (RSVD11[6])—Offset 1F8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.912 Reserved (RSVD11[7])—Offset 1FCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.913 Reserved (RSVD11[8])—Offset 200h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.914 Reserved (RSVD11[9])—Offset 204h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.915 Reserved (RSVD11[10])—Offset 208h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.916 Reserved (RSVD11[11])—Offset 20Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.917 Event Trigger Output Enable (EVOUTEN_0)—Offset 210h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15:0	0h RW	**EV Trigger Output Enable (EVOUTEN):** When a bit is '0', the corresponding output is masked to '0' I.e. no valid pin assigned to. When a bit is set to '1' the corresponding output is enabled and depends on the EVMAP settings.

29.918 Event Trigger Output Enable (EVOUTEN_1)—Offset 214h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15:0	0h RW	**EV Trigger Output Enable (EVOUTEN):** When a bit is '0', the corresponding output is masked to '0' I.e. no valid pin assigned to. When a bit is set to '1' the corresponding output is enabled and depends on the EVMAP settings.

29.919 Event Trigger Mapping (EVMAP_0)—Offset 220h

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RO	**Reserved (RSVD_0):** Reserved
62:60	0h RW	**Event Trigger Mapping to PMU[15] (EVMAPPM_15):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
59	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
58:56	0h RW	**Event Trigger Mapping to PMU[14] (EVMAPPM_14):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
55	0h RO	**Reserved (RSVD_2):** Reserved
54:52	0h RW	**Event Trigger Mapping to PMU[13] (EVMAPPM_13):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
51	0h RO	**Reserved (RSVD_3):** Reserved
50:48	0h RW	**Event Trigger Mapping to PMU[12] (EVMAPPM_12):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
47	0h RO	**Reserved (RSVD_4):** Reserved
46:44	0h RW	**Event Trigger Mapping to PMU[11] (EVMAPPM_11):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
43	0h RO	**Reserved (RSVD_5):** Reserved
42:40	0h RW	**Event Trigger Mapping to PMU[10] (EVMAPPM_10):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
39	0h RO	**Reserved (RSVD_6):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
38:36	0h RW	**Event Trigger Mapping to PMU[9] (EVMAPPM_9):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
35	0h RO	**Reserved (RSVD_7):** Reserved
34:32	0h RW	**Event Trigger Mapping to PMU[8] (EVMAPPM_8):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
31	0h RO	**Reserved (RSVD_8):** Reserved
30:28	0h RW	**Event Trigger Mapping to PMU[7] (EVMAPPM_7):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
27	0h RO	**Reserved (RSVD_9):** Reserved
26:24	0h RW	**Event Trigger Mapping to PMU[6] (EVMAPPM_6):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
23	0h RO	**Reserved (RSVD_10):** Reserved
22:20	0h RW	**Event Trigger Mapping to PMU[5] (EVMAPPM_5):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
19	0h RO	**Reserved (RSVD_11):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
18:16	0h RW	**Event Trigger Mapping to PMU[4] (EVMAPPM_4):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
15	0h RO	**Reserved (RSVD_12):** Reserved
14:12	0h RW	**Event Trigger Mapping to PMU[3] (EVMAPPM_3):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
11	0h RO	**Reserved (RSVD_13):** Reserved
10:8	0h RW	**Event Trigger Mapping to PMU[2] (EVMAPPM_2):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
7	0h RO	**Reserved (RSVD_14):** Reserved
6:4	0h RW	**Event Trigger Mapping to PMU[1] (EVMAPPM_1):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
3	0h RO	**Reserved (RSVD_15):** Reserved
2:0	0h RW	**Event Trigger Mapping to PMU[0] (EVMAPPM_0):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community

29.920 Event Trigger Mapping (EVMAP_1)—Offset 228h

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RO	**Reserved (RSVD_0):** Reserved
62:60	0h RW	**Event Trigger Mapping to PMU[15] (EVMAPPM_15):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
59	0h RO	**Reserved (RSVD_1):** Reserved
58:56	0h RW	**Event Trigger Mapping to PMU[14] (EVMAPPM_14):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
55	0h RO	**Reserved (RSVD_2):** Reserved
54:52	0h RW	**Event Trigger Mapping to PMU[13] (EVMAPPM_13):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
51	0h RO	**Reserved (RSVD_3):** Reserved
50:48	0h RW	**Event Trigger Mapping to PMU[12] (EVMAPPM_12):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
47	0h RO	**Reserved (RSVD_4):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
46:44	0h RW	**Event Trigger Mapping to PMU[11] (EVMAPPM_11):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
43	0h RO	**Reserved (RSVD_5):** Reserved
42:40	0h RW	**Event Trigger Mapping to PMU[10] (EVMAPPM_10):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
39	0h RO	**Reserved (RSVD_6):** Reserved
38:36	0h RW	**Event Trigger Mapping to PMU[9] (EVMAPPM_9):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
35	0h RO	**Reserved (RSVD_7):** Reserved
34:32	0h RW	**Event Trigger Mapping to PMU[8] (EVMAPPM_8):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
31	0h RO	**Reserved (RSVD_8):** Reserved
30:28	0h RW	**Event Trigger Mapping to PMU[7] (EVMAPPM_7):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
27	0h RO	**Reserved (RSVD_9):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
26:24	0h RW	**Event Trigger Mapping to PMU[6] (EVMAPPM_6):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
23	0h RO	**Reserved (RSVD_10):** Reserved
22:20	0h RW	**Event Trigger Mapping to PMU[5] (EVMAPPM_5):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
19	0h RO	**Reserved (RSVD_11):** Reserved
18:16	0h RW	**Event Trigger Mapping to PMU[4] (EVMAPPM_4):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
15	0h RO	**Reserved (RSVD_12):** Reserved
14:12	0h RW	**Event Trigger Mapping to PMU[3] (EVMAPPM_3):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
11	0h RO	**Reserved (RSVD_13):** Reserved
10:8	0h RW	**Event Trigger Mapping to PMU[2] (EVMAPPM_2):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
7	0h RO	**Reserved (RSVD_14):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
6:4	0h RW	**Event Trigger Mapping to PMU[1] (EVMAPPM_1):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community
3	0h RO	**Reserved (RSVD_15):** Reserved
2:0	0h RW	**Event Trigger Mapping to PMU[0] (EVMAPPM_0):** 000b = GPIO[i] event trigger 001b = GPIO[i+16*1] event trigger 010b = GPIO[i+16*2] event trigger 011b = GPIO[i+16*3] event trigger 100b = GPIO[i+16*4] event trigger 101b = GPIO[i+16*5] event trigger 110b = GPIO[i+16*6] event trigger 111b = GPIO[i+16*7] event trigger i = 0, 1, ..., 14, 15 series continues if more than 128 pins per community

29.921 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_JTAGITP_1P8)—Offset 300h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.922 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_JTAGITP_1P8)—Offset 304h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.923 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_JTAGITP_1P8)—Offset 308h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RO	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.924 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_JTAGITP_1P8)—Offset 30Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RO	**strsel B (STRSELB):** Strsel B value

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.925 Family Configuration Register (FAM_CFG_JTAGITP_1P8)—Offset 310h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C00000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RO	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
26	0h RO	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	0h RO	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RO	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	0h RO	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RO	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.926 Family Reserved Register DW5 (FAM_RSVD_DW5JTAGITP_1P8)—Offset 314h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.927 Family Reserved Register DW6 (FAM_RSVD_DW6JTAGITP_1P8)—Offset 318h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.928 Family Reserved Register DW7 (FAM_RSVD_DW7JTAGITP_1P8)—Offset 31Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.929 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_GPHS1_1P8)—Offset 320h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.930 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_GPHS1_1P8)—Offset 324h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.931 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_GPHS1_1P8)—Offset 328h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RO	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.932 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_GPHS1_1P8)—Offset 32Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RO	**strsel B (STRSELB):** Strsel B value
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.933 Family Configuration Register (FAM_CFG_GPHS1_1P8)—Offset 330h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1C03000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RO	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
26	0h RO	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	3h RW	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8

Bit Range	Default & Access	Field Name (ID): Description
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RO	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.934 Family Reserved Register DW5 (FAM_RSVD_DW5GPHS1_1P8)—Offset 334h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.935 Family Reserved Register DW6 (FAM_RSVD_DW6GPHS1_1P8)—Offset 338h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.936 Family Reserved Register DW7 (FAM_RSVD_DW7GPHS1_1P8)—Offset 33Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.937 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_GPHS2_1P8)—Offset 340h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.938 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_GPHS2_1P8)—Offset 344h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.939 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_GPHS2_1P8)—Offset 348h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RO	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.940 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_GPHS2_1P8)—Offset 34Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RO	**strsel B (STRSELB):** Strsel B value
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.941 Family Configuration Register (FAM_CFG_GPHS2_1P8)—Offset 350h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1C03000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RO	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
26	0h RO	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	3h RW	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RO	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.942 Family Reserved Register DW5 (FAM_RSVD_DW5GPHS2_1P8)—Offset 354h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.943 Family Reserved Register DW6 (FAM_RSVD_DW6GPHS2_1P8)—Offset 358h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.944 Family Reserved Register DW7 (FAM_RSVD_DW7GPHS2_1P8)—Offset 35Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.945 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_MISC_1P8)—Offset 360h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.946 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_MISC_1P8)—Offset 364h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.947 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_MISC_1P8)—Offset 368h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RO	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.948 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_MISC_1P8)—Offset 36Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RO	**strsel B (STRSELB):** Strsel B value

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.949 Family Configuration Register (FAM_CFG_MISC_1P8)—Offset 370h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1C00000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RO	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
26	0h RO	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RO	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	0h RO	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RO	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.950 Family Reserved Register DW5 (FAM_RSVD_DW5MISC_1P8)—Offset 374h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.951 Family Reserved Register DW6 (FAM_RSVD_DW6MISC_1P8)—Offset 378h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.952 Family Reserved Register DW7 (FAM_RSVD_DW7MISC_1P8)—Offset 37Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.953 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_I2C_1P8)—Offset 380h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RO	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.954 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_I2C_1P8)—Offset 384h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.955 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_I2C_1P8)—Offset 388h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RO	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.956 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_I2C_1P8)—Offset 38Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RO	**strsel B (STRSELB):** Strsel B value
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.957 Family Configuration Register (FAM_CFG_I2C_1P8)—Offset 390h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1C00007h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RO	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
26	0h RO	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
16	0h RW	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	0h RW	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8

Bit Range	Default & Access	Field Name (ID): Description
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	7h RW	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.958 Family Reserved Register DW5 (FAM_RSVD_DW5I2C_1P8)—Offset 394h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.959 Family Reserved Register DW6 (FAM_RSVD_DW6I2C_1P8)—Offset 398h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.960 Family Reserved Register DW7 (FAM_RSVD_DW7I2C_1P8)—Offset 39Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.961 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_UART_1P8)—Offset 3A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.962 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_UART_1P8)—Offset 3A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.963 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_UART_1P8)—Offset 3A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RO	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.964 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_UART_1P8)—Offset 3ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RO	**strsel B (STRSELB):** Strsel B value
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.965 Family Configuration Register (FAM_CFG_UART_1P8)—Offset 3B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1C00000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RO	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
26	0h RO	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	0h RO	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RO	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.966 Family Reserved Register DW5 (FAM_RSVD_DW5UART_1P8)—Offset 3B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.967 Family Reserved Register DW6 (FAM_RSVD_DW6UART_1P8)—Offset 3B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.968 Family Reserved Register DW7 (FAM_RSVD_DW7UART_1P8)—Offset 3BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.969 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_PMC_1P8)—Offset 3C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.970 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_PMC_1P8)—Offset 3C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.971 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_PMC_1P8)—Offset 3C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RO	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.972 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_PMC_1P8)—Offset 3CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RO	**strsel B (STRSELB):** Strsel B value

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.973 Family Configuration Register (FAM_CFG_PMC_1P8)—Offset 3D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1C00000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RO	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
26	0h RO	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RO	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	0h RO	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RO	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.974 Family Reserved Register DW5 (FAM_RSVD_DW5PMC_1P8)—Offset 3D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.975 Family Reserved Register DW6 (FAM_RSVD_DW6PMC_1P8)—Offset 3D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.976 Family Reserved Register DW7 (FAM_RSVD_DW7PMC_1P8)—Offset 3DCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.977 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_v_CNV0)—Offset 3E0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RO	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.978 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_v_CNV0)—Offset 3E4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.979 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_v_CNV0)—Offset 3E8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RO	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.980 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_v_CNV0)—Offset 3ECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RO	**strsel B (STRSELB):** Strsel B value
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.981 Family Configuration Register (FAM_CFG_v_CNV0)—Offset 3F0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C00000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RO	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
26	0h RO	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	0h RO	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	0h RO	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8

Bit Range	Default & Access	Field Name (ID): Description
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RO	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.982 Family Reserved Register DW5 (FAM_RSVD_DW5v_CNV0)—Offset 3F4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.983 Family Reserved Register DW6 (FAM_RSVD_DW6v_CNV0)—Offset 3F8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.984 Family Reserved Register DW7 (FAM_RSVD_DW7v_CNV0)—Offset 3FCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.985 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_0)—Offset 600h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad

Bit Range	Default & Access	Field Name (ID): Description
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.986 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_0)—Offset 604h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C01000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RW	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	3h RW	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.987 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_0)—Offset 608h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.988 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_0)—Offset 60Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.989 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_1)—Offset 610h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)

Bit Range	Default & Access	Field Name (ID): Description
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.990 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_1)—Offset 614h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C01000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RW	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	3h RW	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.991 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_1)—Offset 618h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.992 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_1)—Offset 61Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.993 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_2)—Offset 620h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.994 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_2)—Offset 624h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C03000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RW	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	3h RW	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.995 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_2)—Offset 628h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.996 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_2)—Offset 62Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.997 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_3)—Offset 630h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.998 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_3)—Offset 634h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C03000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RW	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	3h RW	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.999 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_3)—Offset 638h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1000 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_3)—Offset 63Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1001 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_4)—Offset 640h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1002 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_4)—Offset 644h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C03000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RW	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	3h RW	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1003 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_4)—Offset 648h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1004 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_4)—Offset 64Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1005 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_5)—Offset 650h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1006 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_5)—Offset 654h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C03000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RW	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	3h RW	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1007 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_5)—Offset 658h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1008 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_5)—Offset 65Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1009 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_6)—Offset 660h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1010 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_6)—Offset 664h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C03000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RW	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	3h RW	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1011 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_6)—Offset 668h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1012 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_6)—Offset 66Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1013 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_7)—Offset 670h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1014 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_7)—Offset 674h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C03000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RW	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	3h RW	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1015 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_7)—Offset 678h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1016 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_7)—Offset 67Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1017 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_8)—Offset 680h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1018 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_8)—Offset 684h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1032h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	32h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1019 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_8)—Offset 688h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1020 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_8)—Offset 68Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1021 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_9)—Offset 690h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1022 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_9)—Offset 694h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1033h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	33h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1023 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_9)—Offset 698h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1024 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_9)—Offset 69Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1025 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_10)—Offset 6A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1026 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_10)—Offset 6A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1034h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	34h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1027 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_10)—Offset 6A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1028 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_10)—Offset 6ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1029 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_11)—Offset 6B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1030 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_11)—Offset 6B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1035h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	35h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1031 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_11)—Offset 6B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1032 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_11)—Offset 6BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1033 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_12)—Offset 6C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1034 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_12)—Offset 6C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1036h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	36h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1035 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_12)—Offset 6C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1036 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_12)—Offset 6CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1037 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_13)—Offset 6D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1038 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_13)—Offset 6D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1037h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	37h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1039 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_13)—Offset 6D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1040 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_13)—Offset 6DCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1041 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_14)—Offset 6E0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1042 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_14)—Offset 6E4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1038h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	38h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1043 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_14)—Offset 6E8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1044 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_14)—Offset 6ECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1045 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_15)—Offset 6F0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1046 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_15)—Offset 6F4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1039h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	39h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1047 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_15)—Offset 6F8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1048 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_15)—Offset 6FCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1049 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_16)—Offset 700h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1050 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_16)—Offset 704h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 103Ah

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	3Ah RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1051 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_16)—Offset 708h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1052 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_16)—Offset 70Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1053 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_17)—Offset 710h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1054 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_17)—Offset 714h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 303Bh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	3Bh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1055 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_17)—Offset 718h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1056 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_17)—Offset 71Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1057 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_18)—Offset 720h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1058 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_18)—Offset 724h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 303Ch

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	3Ch RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1059 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_18)—Offset 728h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1060 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_18)—Offset 72Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1061 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_19)—Offset 730h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1062 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_19)—Offset 734h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 303Dh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	3Dh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1063 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_19)—Offset 738h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1064 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_19)—Offset 73Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1065 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_20)—Offset 740h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1066 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_20)—Offset 744h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 303Eh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	3Eh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1067 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_20)—Offset 748h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1068 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_20)—Offset 74Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1069 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_21)—Offset 750h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1070 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_21)—Offset 754h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 303Fh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	3Fh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1071 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_21)—Offset 758h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1072 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_21)—Offset 75Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1073 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_22)—Offset 760h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1074 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_22)—Offset 764h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3040h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	40h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1075 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_22)—Offset 768h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1076 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_22)—Offset 76Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1077 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_23)—Offset 770h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000300h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1078 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_23)—Offset 774h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3041h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	41h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1079 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_23)—Offset 778h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1080 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_23)—Offset 77Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1081 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_24)—Offset 780h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000300h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1082 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_24)—Offset 784h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3042h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	42h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1083 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_24)—Offset 788h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1084 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_24)—Offset 78Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1085 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_25)—Offset 790h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000300h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1086 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_25)—Offset 794h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3043h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	43h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1087 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_25)—Offset 798h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1088 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_25)—Offset 79Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1089 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_26)—Offset 7A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1090 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_26)—Offset 7A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1044h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	44h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1091 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_26)—Offset 7A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1092 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_26)—Offset 7ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1093 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_27)—Offset 7B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1094 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_27)—Offset 7B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1045h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	45h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1095 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_27)—Offset 7B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 800h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	1h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1096 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_27)—Offset 7BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1097 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_28)—Offset 7C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1098 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_28)—Offset 7C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1046h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	46h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1099 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_28)—Offset 7C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 800h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	1h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1100 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_28)—Offset 7CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1101 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_29)—Offset 7D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1102 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_29)—Offset 7D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1047h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	47h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1103 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_29)—Offset 7D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1104 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_29)—Offset 7DCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1105 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_30)—Offset 7E0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1106 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_30)—Offset 7E4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1048h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	48h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1107 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_30)—Offset 7E8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1108 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_30)—Offset 7ECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1109 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_31)—Offset 7F0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1110 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_31)—Offset 7F4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1049h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	49h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1111 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_31)—Offset 7F8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1112 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_31)—Offset 7FCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1113 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_32)—Offset 800h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1114 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_32)—Offset 804h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 104Ah

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	4Ah RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1115 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_32)—Offset 808h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1116 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_32)—Offset 80Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1117 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_33)—Offset 810h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1118 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_33)—Offset 814h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 104Bh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	4Bh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1119 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_33)—Offset 818h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1120 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_33)—Offset 81Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1121 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_34)—Offset 820h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1122 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_34)—Offset 824h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 104Ch

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	4Ch RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1123 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_34)—Offset 828h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1124 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_34)—Offset 82Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1125 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_35)—Offset 830h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44001B00h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	6h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1126 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_35)—Offset 834h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 304Dh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	4Dh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1127 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_35)—Offset 838h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1128 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_35)—Offset 83Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1129 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_36)—Offset 840h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44001B00h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	6h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1130 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_36)—Offset 844h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 304Eh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	4Eh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1131 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_36)—Offset 848h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1132 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_36)—Offset 84Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1133 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_37)—Offset 850h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1134 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_37)—Offset 854h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 304Fh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	4Fh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1135 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_37)—Offset 858h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1136 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_37)—Offset 85Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1137 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_38)—Offset 860h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1138 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_38)—Offset 864h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3050h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	50h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1139 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_38)—Offset 868h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1140 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_38)—Offset 86Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1141 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_39)—Offset 870h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1142 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_39)—Offset 874h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3051h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	51h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1143 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_39)—Offset 878h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1144 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_39)—Offset 87Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1145 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_40)—Offset 880h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1146 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_40)—Offset 884h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3052h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	52h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1147 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_40)—Offset 888h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1148 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_40)—Offset 88Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1149 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_41)—Offset 890h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1150 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_41)—Offset 894h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1053h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	53h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1151 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_41)—Offset 898h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1152 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_41)—Offset 89Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1153 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_42)—Offset 8A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1154 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_42)—Offset 8A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1054h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	54h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1155 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_42)—Offset 8A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1156 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_42)—Offset 8ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1157 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_43)—Offset 8B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:13	0h RO	**Reserved (RSVD_1):** Reserved
12:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1158 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_43)—Offset 8B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1055h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	55h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1159 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_43)—Offset 8B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 800h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	1h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1160 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_43)—Offset 8BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1161 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_44)—Offset 8C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:13	0h RO	**Reserved (RSVD_1):** Reserved
12:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1162 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_44)—Offset 8C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3056h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	56h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1163 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_44)—Offset 8C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1164 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_44)—Offset 8CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1165 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_45)—Offset 8D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1166 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_45)—Offset 8D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3057h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	57h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1167 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_45)—Offset 8D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1168 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_45)—Offset 8DCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1169 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_46)—Offset 8E0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1170 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_46)—Offset 8E4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1058h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	58h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1171 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_46)—Offset 8E8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1172 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_46)—Offset 8ECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1173 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_47)—Offset 8F0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1174 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_47)—Offset 8F4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1059h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	59h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1175 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_47)—Offset 8F8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1176 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_47)—Offset 8FCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1177 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_48)—Offset 900h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1178 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_48)—Offset 904h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 245Ah

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	9h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	5Ah RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1179 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_48)—Offset 908h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1180 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_48)—Offset 90Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1181 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_49)—Offset 910h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1182 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_49)—Offset 914h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 245Bh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	9h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	5Bh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1183 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_49)—Offset 918h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1184 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_49)—Offset 91Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1185 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_50)—Offset 920h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1186 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_50)—Offset 924h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 305Ch

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	5Ch RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1187 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_50)—Offset 928h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1188 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_50)—Offset 92Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1189 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_51)—Offset 930h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1190 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_51)—Offset 934h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 305Dh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	5Dh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1191 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_51)—Offset 938h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1192 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_51)—Offset 93Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1193 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_52)—Offset 940h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1194 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_52)—Offset 944h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 305Eh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	5Eh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1195 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_52)—Offset 948h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1196 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_52)—Offset 94Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1197 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_53)—Offset 950h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1198 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_53)—Offset 954h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 305Fh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	5Fh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1199 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_53)—Offset 958h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1200 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_53)—Offset 95Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1201 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_54)—Offset 960h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1202 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_54)—Offset 964h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3060h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	60h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1203 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_54)—Offset 968h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1204 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_54)—Offset 96Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1205 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_55)—Offset 970h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1206 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_55)—Offset 974h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3061h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	61h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1207 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_55)—Offset 978h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1208 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_55)—Offset 97Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1209 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_56)—Offset 980h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1210 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_56)—Offset 984h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3062h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	62h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1211 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_56)—Offset 988h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1212 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_56)—Offset 98Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1213 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_57)—Offset 990h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1214 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_57)—Offset 994h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3063h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	63h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1215 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_57)—Offset 998h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1216 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_57)—Offset 99Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1217 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_58)—Offset 9A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1218 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_58)—Offset 9A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3064h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	64h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1219 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_58)—Offset 9A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1220 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_58)—Offset 9ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1221 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_59)—Offset 9B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1222 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_59)—Offset 9B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3065h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RW	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	65h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1223 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_59)—Offset 9B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1224 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_59)—Offset 9BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1225 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_60)—Offset 9C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1226 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_60)—Offset 9C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3066h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	66h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1227 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_60)—Offset 9C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1228 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_60)—Offset 9CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1229 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_61)—Offset 9D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000300h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1230 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_61)—Offset 9D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3067h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	67h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1231 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_61)—Offset 9D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1232 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_61)—Offset 9DCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1233 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_62)—Offset 9E0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000300h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1234 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_62)—Offset 9E4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3068h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	68h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1235 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_62)—Offset 9E8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1236 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_62)—Offset 9ECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1237 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_63)—Offset 9F0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1238 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_63)—Offset 9F4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3069h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	69h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1239 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_63)—Offset 9F8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1240 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_63)—Offset 9FCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1241 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_64)—Offset A00h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1242 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_64)—Offset A04h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 306Ah

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	6Ah RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1243 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_64)—Offset A08h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1244 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_64)—Offset A0Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1245 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_65)—Offset A10h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000300h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1246 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_65)—Offset A14h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 306Bh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	6Bh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1247 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_65)—Offset A18h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1248 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_65)—Offset A1Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1249 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_66)—Offset A20h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000300h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1250 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_66)—Offset A24h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 306Ch

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	6Ch RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1251 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_66)—Offset A28h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1252 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_66)—Offset A2Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1253 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_67)—Offset A30h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1254 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_67)—Offset A34h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 306Dh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	6Dh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1255 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_67)—Offset A38h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1256 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_67)—Offset A3Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1257 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_68)—Offset A40h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1258 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_68)—Offset A44h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 306Eh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	6Eh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1259 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_68)—Offset A48h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1260 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_68)—Offset A4Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1261 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_69)—Offset A50h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1262 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_69)—Offset A54h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 306Fh

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	6Fh RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1263 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_69)—Offset A58h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1264 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_69)—Offset A5Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1265 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_70)—Offset A60h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1266 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_70)—Offset A64h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3070h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	70h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1267 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_70)—Offset A68h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1268 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_70)—Offset A6Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1269 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_71)—Offset A70h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1270 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_71)—Offset A74h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1071h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	71h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1271 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_71)—Offset A78h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1272 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_71)—Offset A7Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1273 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_72)—Offset A80h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1274 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_72)—Offset A84h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1072h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	72h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1275 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_72)—Offset A88h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1276 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_72)—Offset A8Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1277 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_73)—Offset A90h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1278 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_73)—Offset A94h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1073h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	73h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1279 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_73)—Offset A98h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1280 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_73)—Offset A9Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1281 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_74)—Offset AA0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1282 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_74)—Offset AA4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1283 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_74)—Offset AA8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1284 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_74)—Offset AACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1285 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_75)—Offset AB0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1286 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_75)—Offset AB4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1287 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_75)—Offset AB8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1288 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_75)—Offset ABCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1289 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_211)—Offset AC0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1290 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_211)—Offset AC4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3074h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	74h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1291 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_211)—Offset AC8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1292 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_211)—Offset ACCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1293 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_212)—Offset AD0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RO	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1294 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_212)—Offset AD4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3075h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	75h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1295 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_212)—Offset AD8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1296 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_212)—Offset ADCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1297 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_213)—Offset AE0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RO	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1298 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_213)—Offset AE4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1076h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	76h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1299 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_213)—Offset AE8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1300 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_213)—Offset AECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1301 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_214)—Offset AF0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RO	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1302 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_214)—Offset AF4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1077h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	77h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1303 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_214)—Offset AF8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1304 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_214)—Offset AFCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1305 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_0)—Offset B00h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000201h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RO	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	0h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	1h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1306 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_0)—Offset B04h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1307 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_0)—Offset B08h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1308 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_0)—Offset B0Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1309 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_1)—Offset B10h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000200h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RO	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	0h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1310 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_1)—Offset B14h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1311 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_1)—Offset B18h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1312 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_1)—Offset B1Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1313 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_2)—Offset B20h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000200h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RO	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	0h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1314 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_2)—Offset B24h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1315 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_2)—Offset B28h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1316 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_2)—Offset B2Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1317 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_3)—Offset B30h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000200h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RO	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	0h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1318 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_3)—Offset B34h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1319 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_3)—Offset B38h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1320 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_3)—Offset B3Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1321 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_4)—Offset B40h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RO	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1322 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_4)—Offset B44h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 76h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	76h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1323 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_4)—Offset B48h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1324 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_4)—Offset B4Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1325 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_5)—Offset B50h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000201h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RO	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	0h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	1h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1326 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_5)—Offset B54h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1327 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_5)—Offset B58h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1328 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_5)—Offset B5Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1329 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_6)—Offset B60h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1330 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_6)—Offset B64h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1331 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_6)—Offset B68h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1332 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_6)—Offset B6Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1333 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_7)—Offset B70h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1334 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_7)—Offset B74h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1335 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_7)—Offset B78h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1336 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_7)—Offset B7Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1337 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_8)—Offset B80h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1338 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_8)—Offset B84h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1339 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_8)—Offset B88h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1340 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_8)—Offset B8Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1341 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_9)—Offset B90h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1342 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_9)—Offset B94h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1343 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_9)—Offset B98h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1344 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_9)—Offset B9Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1345 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_10)—Offset BA0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000B00h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	2h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1346 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_10)—Offset BA4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1347 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_10)—Offset BA8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1348 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_10)—Offset BACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1349 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_11)—Offset BB0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000B00h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	2h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1350 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_11)—Offset BB4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1351 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_11)—Offset BB8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1352 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_11)—Offset BBCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1353 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_12)—Offset BC0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000B00h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	2h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1354 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_12)—Offset BC4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1355 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_12)—Offset BC8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1356 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_12)—Offset BCCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1357 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_13)—Offset BD0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000B00h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	2h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1358 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_13)—Offset BD4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1359 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_13)—Offset BD8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1360 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_13)—Offset BDCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1361 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_14)—Offset BE0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1362 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_14)—Offset BE4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1363 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_14)—Offset BE8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1364 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_14)—Offset BECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1365 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_15)—Offset BF0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1366 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_15)—Offset BF4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1367 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_15)—Offset BF8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1368 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_15)—Offset BFCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1369 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_16)—Offset C00h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1370 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_16)—Offset C04h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1371 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_16)—Offset C08h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1372 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_16)—Offset C0Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1373 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_17)—Offset C10h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1374 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_17)—Offset C14h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1375 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_17)—Offset C18h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1376 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_17)—Offset C1Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1377 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_18)—Offset C20h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1378 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_18)—Offset C24h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1379 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_18)—Offset C28h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1380 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_18)—Offset C2Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1381 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_19)—Offset C30h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1382 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_19)—Offset C34h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1383 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_19)—Offset C38h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1384 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_19)—Offset C3Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1385 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_20)—Offset C40h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1386 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_20)—Offset C44h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1387 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_20)—Offset C48h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1388 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_20)—Offset C4Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1389 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_21)—Offset C50h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1390 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_21)—Offset C54h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1391 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_21)—Offset C58h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1392 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_21)—Offset C5Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1393 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_22)—Offset C60h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1394 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_22)—Offset C64h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1395 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_22)—Offset C68h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1396 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_22)—Offset C6Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1397 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_23)—Offset C70h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1398 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_23)—Offset C74h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1399 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_23)—Offset C78h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1400 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_23)—Offset C7Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1401 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_24)—Offset C80h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1402 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_24)—Offset C84h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1403 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_24)—Offset C88h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1404 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_24)—Offset C8Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1405 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_25)—Offset C90h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1406 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_25)—Offset C94h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1407 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_25)—Offset C98h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1408 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_25)—Offset C9Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1409 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_26)—Offset CA0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1410 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_26)—Offset CA4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1411 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_26)—Offset CA8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1412 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_26)—Offset CACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1413 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_27)—Offset CB0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1414 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_27)—Offset CB4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1415 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_27)—Offset CB8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1416 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_27)—Offset CBCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1417 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_28)—Offset CC0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1418 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_28)—Offset CC4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1419 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_28)—Offset CC8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1420 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_28)—Offset CCCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1421 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_29)—Offset CD0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1422 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_29)—Offset CD4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1423 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_29)—Offset CD8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1424 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_29)—Offset CDCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1425 Pad Configuration DW0 (PAD_CFG_DW0_vGPIO_30)—Offset CE0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RO	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1426 Pad Configuration DW1 (PAD_CFG_DW1_vGPIO_30)—Offset CE4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 77h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RO	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RO	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	77h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1427 Pad Configuration DW2 (PAD_CFG_DW2_vGPIO_30)—Offset CE8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RO	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RO	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1428 Pad Configuration DW3 (PAD_CFG_DW3_vGPIO_30)—Offset CECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

Revision ID (REV_ID)—Offset 0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 940000h

Bit Range	Default & Access	Field Name (ID): Description
31:16	94h RO	**Hardware Base Revision ID (HWBASEREVID):** This field is used to indicate the baseline hardware revision. 0091h = RTL release for HAS0.91

Bit Range	Default & Access	Field Name (ID): Description
15:0	0h RO	**Hardware Sub Revision ID (HWSUBIREVID):** This field is used to indicate the revision of the hardware derived from the baseline.

29.1429 Capability List Register (CAP_LIST_0)—Offset 4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RO	**Reserved (RSVD_0):** Reserved
23:16	0h RO	**Capability Identification (CAPID):** A unique identification for the current capability. A value of 0 is reserved, and must not be used by any capability. Note:This field is always 0 for the first Capability List register.
15:0	0h RO	**Next Capability List Pointer (NXTCAPLPTR):** Specify the DW-aligned Pointer/ Address to the next item in this capabilities list and must be 0 if there is no capability at all or this is the last capability in the list.

29.1430 Family Base Address (FAMBAR)—Offset 8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 300h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15:0	300h RO	**Family Base Address (FAMBAR):** This field provides the starting byte-align address of Family0 register sets within a Community. It is meant for software to discover from where the very first Family register (i.e. Family0 register) starts to compute the next Families address offsets.

29.1431 Pad Base Address (PADBAR)—Offset Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 600h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Reserved (RSVD_0):** Reserved
15:0	600h RO	**Pad Base Address (PADBAR):** This field provides the starting byte-align address of Pad0 register sets within a Community. It is meant for software to discover from where the very first Pad register (i.e. Pad0 register) starts to compute the next Pad address offsets.

29.1432 Miscellaneous Configuration (MISCCFG)—Offset 10h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: E043200h

Bit Range	Default & Access	Field Name (ID): Description
31:24	Eh RW	**GPIO Driver Mode Interrupt Select (GPDMINTSEL):** IRQ globally for all pads (GPI_IS with corresponding GPI_IE enable). 0 = Interrupt Line 0 1 = Interrupt Line 1 ... 255 = Interrupt Line 255
23:20	0h RW	**Reserved (Reserved1):** Reserved
19:16	4h RW	**GPIO Group to GPE_DW2 assignment encoding (GPE0_DW2):** This register assigns a specific GPIO Group to the ACPI GPE0[95:64]. The setting is SoC specific.
15:12	3h RW	**GPIO Group to GPE_DW1 assignment encoding (GPE0_DW1):** This register assigns a specific GPIO Group to the ACPI GPE0[63:32]. The setting is SoC specific.
11:8	2h RW	**GPIO Group to GPE_DW0 assignment encoding (GPE0_DW0):** This register assigns a specific GPIO Group to the ACPI GPE0[31:0]. The setting is SoC specific.
7:6	0h RW	**Reserved (Reserved2):** Reserved
5	0h RW	**GPIO IOSF Sideband Dynamic Partition Clock Gating Enable (GPSIDEDPCGEN):** Specify whether the GPIO Community IOSF sideband clock should take part in partition clock gating 0 = Disable participation in IOSF sideband clock dynamic partition clock gating 1 = Enable participation in IOSF sideband clock dynamic partition clock gating
4	0h RW	**GPIO RCOMP clock Dynamic Local Clock Gating (GPRCOMPCDLCGEN):** Specify whether the GPIO Community should perform local clock gating on RCOMP clock 0 = Disable dynamic RCOMP clock local clock gating 1 = Enable dynamic RCOMP clock local clock gating
3	0h RW	**GPIO RTC clock Dynamic Local Clock Gating (GPRTCDLCGEN):** Specify whether the GPIO Community should perform local clock gating on RTC clock 0 = Disable dynamic RTC clock local clock gating 1 = Enable dynamic RTC clock local clock gating
2	0h RW	**GSX Static Local Clock Gating (GSXSLCGEN):** Specify whether the GSX controller should be statically clock gated for power saving if it is not enabled (even though the capability is available). 0 = Disable static local clock gating 1 = Enable static local clock gating
1	0h RW	**GPIO Dynamic Partition Clock Gating Enable (GPDPCGEN):** Specify whether the GPIO Community should take part in partition clock gating 0 = Disable participation in dynamic partition clock gating 1 = Enable participation in dynamic partition clock gating

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**GPIO Dynamic Local Clock Gating Enable (GPDLCGEN):** Specify whether the GPIO Community should perform local clock gating 0 = Disable dynamic local clock gating 1 = Enable dynamic local clock gating

29.1433 Miscellaneous Secured Configuration (MISCSECCFG)—Offset 14h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD_0):** Reserved
7:4	0h RW	**Reserved (Reserved1):** Reserved
3	0h RW/1C/V	**Soft strap pull status bit (SSTRAPPULLSTAT):** Indicate the status of the last soft strap pull operation. If the implementation guarantee the success of the operation, then this bit can be hard-wired to 0. 0 = No error 1 = Error FW writes 1 to this bit to clear. This bit is set to 1 when soft strap pull request has been completed with completion without data (for both successful and unsuccessful completion status) or completion with data but with unsuccessful completion status. This bit will also set to 1 if there is any failure on the security and error checking in the fuse puller such as invalid SAI and etc. Fuse puller will assert puller_error in this case and this will set this bit to 1.
2	0h RW/1C/V	**Soft strap re-pull done bit (SSTRAPPULLDONE):** Indication if the soft strap pull has finished the operation 0 = operation is not done. Either not started or in progress 1 = operation is done/finish. FW writes 1 to this bit to clear.
1	0h RW	**Soft strap re-pull request bit (SSTRAPREPULLREQ):** Specify if fuse puller need to re-pull the soft strap. The SoC/PMC/FW can set this bit to request the community fuse puller to re-pull the soft strap. 0 to 1 transition = soft strap re-pull request trigger. A soft strap re-pull request will not trigger IP_READY message sent as the IP_READY message shall be sent during the initial soft strap pull by HW as part of the reset sequences. The SSTRAPREPULLREQ, SSTRAPPULLDONE and SSTRAPPULLSTAT are only meaningful for community with soft strap pull capability.
0	0h RW	**Event Trigger IOSF-SB Message Initiation Disable (SBTRIGDIS):** This bit disables the following list of IOSF-SB Message initiation. It does not disable the event trigger. When this bit is cleared to '0', any pending message due to event trigger while this bit is '1' shall be sent on IOSF-SB. Event list: 'Virtual Wire' message IRQ GPE/SCI NMI SMI

29.1434 Reserved (RSVD0[0])—Offset 18h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1435 Reserved (RSVD0[1])—Offset 1Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1436 Pad Ownership (PAD_OWN_scc_0)—Offset 20h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_182):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_181):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_180):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_179):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_187):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_178):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_177):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_176):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.1437 Pad Ownership (PAD_OWN_scc_1)—Offset 24h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_190):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_189):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_210):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_188):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_186):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_185):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_184):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_183):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.1438 Pad Ownership (PAD_OWN_scc_2)—Offset 28h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_198):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_197):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_196):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_195):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_194):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_193):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_192):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_191):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.1439 Pad Ownership (PAD_OWN_scc_3)—Offset 2Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:28	0h RW	**Pad Ownership (PAD_OWN_GPIO_206):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
27:26	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
25:24	0h RW	**Pad Ownership (PAD_OWN_GPIO_205):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
23:22	0h RO	**Reserved (RSVD_2):** Reserved
21:20	0h RW	**Pad Ownership (PAD_OWN_GPIO_204):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
19:18	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
17:16	0h RW	**Pad Ownership (PAD_OWN_GPIO_203):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
15:14	0h RO	**Reserved (RSVD_4):** Reserved
13:12	0h RW	**Pad Ownership (PAD_OWN_GPIO_202):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/ write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
11:10	0h RO	**Reserved (RSVD_5):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_201):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_6):** Reserved
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_200):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_7):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_199):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.1440 Pad Ownership (PAD_OWN_scc_4)—Offset 30h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RW	**Pad Ownership (PAD_OWN_GPIO_209):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
7:6	0h RO	**Reserved (RSVD_1):** Reserved
5:4	0h RW	**Pad Ownership (PAD_OWN_GPIO_208):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.
3:2	0h RO	**Reserved (RSVD_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**Pad Ownership (PAD_OWN_GPIO_207):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. Status update: o GPIO input event notification (if enabled) to embedded system owner via IOSF-SB Virtual Wire message, or o Host SMI (if implemented), NMI (if implemented), SCI/GPE or GPI_INT status update All subsequent read/write accesses to the corresponding pad's Pad Configuration Register are verified with the owner's GPIO Function SAI. PAD_OWN[1:0] encoding: '00' = Host GPIO ACPI Mode or GPIO Driver Mode. Host software (ACPI or GPIO Driver) has ownership of the pad. In Host GPIO Driver Mode (refer to HOSTSW_OWN), GPIO input event update is limited to GPI_STS update only. Otherwise in Host ACPI Mode, updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. No read/write restriction to the Pad Configuration register set during host ownership. During host ownership, CSME, IE and ISH do not own this pad and are not notified of the GPIO input event. '01' = CSME GPIO Mode. CSME has ownership of the pad. All accesses to Pad Configuration register set must be verified with CSME GPIO Function SAI **. Non-posted cycle access without CSME SAI shall result into nsuccessful Completionstatus. Posted cycle access without CSME SAI shall be dropped. In CSME GPIO Mode, the host related status registers related to the pad (refer to host ownership above) are masked from being updated by GPIO input event. If IOSF-SB Virtual Wire message generation is enabled for this pad, GPIO hardware sends the Virtual Wire message to CSME GPIO Function. Only CSME GPIO Function is notified of the GPIO input event. '10' = ISH GPIO Mode. Same description as CSME GPIO Mode above except verifying ISH GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. '11' = IE GPIO Mode. Same description as CSME GPIO Mode above except verifying IE GPIO Function SAI for pad related register accesses and the Virtual Wire message is delivered to the ISH GPIO Function. All other values are reserved. Implementation shall treat reserved values the same as 0h. ** During DFx mode, Secure DFx can write to the PAD_OWN.

29.1441 Reserved (RSVD1[0])—Offset 34h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1442 Reserved (RSVD1[1])—Offset 38h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1443 Reserved (RSVD1[2])—Offset 3Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1444 Reserved (RSVD1[3])—Offset 40h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1445 Reserved (RSVD1[4])—Offset 44h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1446 Reserved (RSVD1[5])—Offset 48h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1447 Reserved (RSVD1[6])—Offset 4Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1448 Reserved (RSVD1[7])—Offset 50h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1449 Reserved (RSVD1[8])—Offset 54h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1450 Reserved (RSVD1[9])—Offset 58h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1451 Reserved (RSVD1[10])—Offset 5Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1452 GPI Virtual Wire Message Enable (GPI_VWE_scc_0)—Offset 60h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_206):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
30	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_205):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
29	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_204):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
28	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_203):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_202):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
26	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_201):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
25	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_200):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
24	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_199):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
23	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_198):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
22	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_197):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
21	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_196):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
20	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_195):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
19	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_194):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
18	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_193):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
17	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_192):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
16	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_191):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
15	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_190):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
14	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_189):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_210):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_188):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
11	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_186):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_185):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
9	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_184):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_183):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_182):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_181):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
5	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_180):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_179):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_187):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_178):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_177):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_176):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.1453 GPI Virtual Wire Message Enable (GPI_VWE_scc_1)—Offset 64h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RO	**Reserved (RSVD_0):** Reserved
2	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_209):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_208):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RO	**GPI Virtual Wire Message Enable (GPI_VWE_GPIO_207):** This register can be written by CSME GPIO Function** only. Write access without CSME GPIO Function SAI shall be dropped. There is no SAI restriction on register read. This bit is used to enable the generation of IOSF-SB Virtual Wire (VW) message corresponding to pad input event. When either an edge or a level event is detected (refer to RxEdCfg, RxInv) on pad configured to GPIO Mode (PMode) input and the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN), a VM message is delivered to the pad owner. This only has effect when the corresponding pad is under CSME, ISH or IE ownership (PAD_OWN). 0 = disable VM message generation 1 = enable VM message generation Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.1454 Reserved (RSVD2[0])—Offset 68h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1455 Reserved (RSVD2[1])—Offset 6Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1456 Reserved (RSVD2[2])—Offset 70h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1457 Reserved (RSVD2[3])—Offset 74h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1458 Reserved (RSVD2[4])—Offset 78h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1459 Reserved (RSVD2[5])—Offset 7Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1460 Pad Configuration Lock (PADCFGLOCK_scc_0)—Offset 80h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_206):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
30	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_205):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
29	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_204):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
28	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_203):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_202):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
26	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_201):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_200):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
24	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_199):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_198):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
22	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_197):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_196):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
20	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_195):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_194):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
18	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_193):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_192):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
16	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_191):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_190):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
14	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_189):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_210):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
12	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_188):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_186):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
10	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_185):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_184):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
8	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_183):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_182):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
6	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_181):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_180):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
4	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_179):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_187):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
2	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_178):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_177):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
0	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_176):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.1461 Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_scc_0)—Offset 84h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_206):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
30	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_205):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
29	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_204):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
28	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_203):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_202):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
26	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_201):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_200):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
24	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_199):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_198):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
22	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_197):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_196):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
20	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_195):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_194):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
18	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_193):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_192):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
16	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_191):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_190):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
14	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_189):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_210):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
12	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_188):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_186):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
10	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_185):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_184):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
8	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_183):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_182):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
6	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_181):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_180):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
4	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_179):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_187):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
2	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_178):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_177):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
0	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_176):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.1462 Pad Configuration Lock (PADCFGLOCK_scc_1)—Offset 88h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_209):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
1	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_208):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**Pad Config Lock (PADCFGLOCK_GPIO_207):** PadCfgLock locks specific register fields in the pad specific registers (in Community or Pad) from being configured. The registers affected become Read-Only and software writes to these registers have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Lock the following register fields as read-only (RO): Pad Configuration registers (exclude GPIOTXState) GPI_NMI_EN Register (if implemented) GPI_SMI_EN Register (if implemented) GPI_GPE_EN Register (if implemented) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 When PadCfgLock[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted cycle that is end point specific to GPIO Community. The Opcode shall be 13h. Posted access has no effect and shall be dropped internally. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific register access message PIO_PadCfg_Unlockwith Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h and Posted='0'. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.1463 Pad Configuration Lock GPIO TxState Register (PADCFGLOCKTX_scc_1)—Offset 8Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_209):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.
1	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_208):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**Pad Config Lock TXState (PADCFGLOCKTX_GPIO_207):** PadCfgLockTx locks the GPIOTxState bit from being configured. The GPIOTxState register becomes Read-Only and software writes to the register have no effect. This field is meaningful and takes effect only if the pad is owned by host (refer to PAD_OWN register). Pads under other ownership are not affected by the PadCfgLock. 0 = Unlock 1 = Locks the Pad Configuration GPIOTXState field as read-only (RO) Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1 = pad N-1 When PadCfgLockTx[n] is written from a '1' to a '0' (unlock), a synchronous SMI# is generated if enabled. This ensures that only SMM code can change the above GPIO registers after the lockdown. In order to support this optional mechanism, this register access type has to be RW and access with non-posted write cycle that is end point specific to GPIO Community. The Opcode for the write shall be 13h. Write cycle with any other opcode has no effect on the PadCfgLockTx register value. Note to BIOS: The non-posted write cycle can be generated using the P2SB bridge's ideband Message Interface Accessmethod in order to trigger a GPIO end point specific IOSF-SB message PIO_PadCfg_Unlock with Opcode of 13h. This can be achieved by programming the P2SB configuration register SBISTAT.Opcode to 13h. Refer to P2SB Cspec chapter ideband Message Interface Accesssection.

29.1464 Reserved (RSVD3[0])—Offset 90h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1465 Reserved (RSVD3[1])—Offset 94h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1466 Reserved (RSVD3[2])—Offset 98h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1467 Reserved (RSVD3[3])—Offset 9Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1468 Reserved (RSVD3[4])—Offset A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1469 Reserved (RSVD3[5])—Offset A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1470 Reserved (RSVD3[6])—Offset A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1471 Reserved (RSVD3[7])—Offset ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1472 Host Software Pad Ownership (HOSTSW_OWN_scc_0)—Offset B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_206):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
30	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_205):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
29	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_204):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
28	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_203):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
27	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_202):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_201):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
25	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_200):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
24	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_199):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
23	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_198):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
22	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_197):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_196):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
20	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_195):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
19	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_194):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
18	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_193):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
17	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_192):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_191):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
15	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_190):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
14	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_189):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_210):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_188):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_186):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_185):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
9	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_184):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_183):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_182):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
6	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_181):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_180):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_179):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_187):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_178):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_177):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_176):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.1473 Host Software Pad Ownership (HOSTSW_OWN_scc_1)—Offset B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RO	**Reserved (RSVD_0):** Reserved
2	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_209):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_208):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW	**HostSW_Own (HOSTSW_OWN_GPIO_207):** This register determines the appropriate host status bit update when a pad is under host ownership (refer to PAD_OWN). 0 = ACPI Mode. GPIO input event updates are limited to GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS. GPI_STS update is masked. 1 = GPIO Driver Mode. GPIO input event updates are limited to GPI_STS. GPI_GPE_STS, GPI_NMI_STS and/or GPI_SMI_STS updates are masked. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.1474 Reserved (RSVD4[0])—Offset B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1475 Reserved (RSVD4[1])—Offset BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1476 Reserved (RSVD4[2])—Offset C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1477 Reserved (RSVD4[3])—Offset C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1478 Reserved (RSVD4[4])—Offset C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1479 Reserved (RSVD4[5])—Offset CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1480 Reserved (RSVD4[6])—Offset D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1481 Reserved (RSVD4[7])—Offset D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1482 Reserved (RSVD4[8])—Offset D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1483 Reserved (RSVD4[9])—Offset DCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1484 Reserved (RSVD4[10])—Offset E0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1485 Reserved (RSVD4[11])—Offset E4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1486 Reserved (RSVD4[12])—Offset E8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1487 Reserved (RSVD4[13])—Offset ECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1488 Reserved (RSVD4[14])—Offset F0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1489 Reserved (RSVD4[15])—Offset F4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1490 Reserved (RSVD4[16])—Offset F8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1491 Reserved (RSVD4[17])—Offset FCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1492 GPI Interrupt Status (GPI_IS_scc_0)—Offset 100h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_206):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
30	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_205):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
29	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_204):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
28	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_203):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
27	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_202):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
26	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_201):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_200):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
24	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_199):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
23	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_198):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
22	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_197):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_196):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
20	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_195):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
19	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_194):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
18	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_193):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_192):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
16	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_191):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
15	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_190):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
14	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_189):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_210):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_188):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_186):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_185):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_184):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_183):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_182):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
6	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_181):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_180):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_179):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
3	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_187):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_178):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_177):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_176):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.1493 GPI Interrupt Status (GPI_IS_scc_1)—Offset 104h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RO	**Reserved (RSVD_0):** Reserved
2	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_209):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_208):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW/1C	**GPI Interrupt Status (GPI_INT_STS_GPIO_207):** This bit is set to '1' by hardware when either an edge or a level event is detected (See RxEdCfg RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode The corresponding PAD_OWN[1:0] is '00' and HOSTSW_OWN ='1' (i.e. Host GPIO Driver Mode). Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = No interrupt 1 = Interrupt asserts The state of GPI_INT_EN[x] does not prevent the setting of GPI_INT_STS[x]. Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.1494 Reserved (RSVD5[0])—Offset 108h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1495 Reserved (RSVD5[1])—Offset 10Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1496 GPI Interrupt Enable (GPI_IE_scc_0)—Offset 110h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_206):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
30	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_205):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
29	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_204):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
28	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_203):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_202):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
26	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_201):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
25	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_200):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
24	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_199):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
23	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_198):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
22	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_197):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_196):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
20	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_195):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
19	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_194):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
18	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_193):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
17	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_192):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
16	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_191):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_190):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
14	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_189):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
13	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_210):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
12	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_188):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
11	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_186):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
10	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_185):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_184):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
8	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_183):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
7	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_182):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
6	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_181):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
5	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_180):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
4	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_179):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_187):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
2	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_178):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_177):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_176):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.1497 GPI Interrupt Enable (GPI_IE_scc_1)—Offset 114h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_209):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
1	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_208):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1
0	0h RW	**GPI Interrupt Enable (GPI_INT_EN_GPIO_207):** This bit is used to enable/disable the generation of APIC interrupt when the corresponding GPI_INT_STS bit is set. 0 = disable interrupt generation 1 = enable interrupt generation Refer to GPI_IRQ_ROUTE for host GPIO Driver Mode interrupt routing. Each bit position correspond to 1 pad: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1

29.1498 Reserved (RSVD6[0])—Offset 118h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1499 Reserved (RSVD6[1])—Offset 11Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1500 Reserved (RSVD6[2])—Offset 120h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1501 Reserved (RSVD6[3])—Offset 124h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1502 Reserved (RSVD6[4])—Offset 128h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1503 Reserved (RSVD6[5])—Offset 12Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1504 GPI General Purpose Events Status (GPI_GPE_STS_scc_0)—Offset 130h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_206):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
30	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_205):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
29	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_204):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
28	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_203):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
27	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_202):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
26	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_201):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
25	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_200):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
24	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_199):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
23	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_198):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
22	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_197):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_196):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
20	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_195):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
19	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_194):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
18	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_193):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_192):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
16	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_191):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
15	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_190):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
14	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_189):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_210):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_188):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_186):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_185):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_184):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_183):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_182):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_181):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
5	0h RO	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_180):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
4	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_179):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_187):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_178):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_177):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_176):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.1505 GPI General Purpose Events Status (GPI_GPE_STS_scc_1)—Offset 134h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_209):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_208):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RW/1C	**GPI General Purpose Events Status (GPI_GPE_STS_GPIO_207):** These bits are set any time the corresponding GPIO pad is set up as an input, under host ownership and the corresponding GPIO signal is high (or low if the corresponding RXINV bit is set). If the corresponding enable bit is set in the GPI_GPE_EN register, then when the GPI_GPE_STS[i] bit is set: If the system is in an S3-S5 state, the event will also wake the system. Refer to Cspec Chapter 16 Power Management. If the system is in an S0 state (or upon waking back to an S0 state), an SCI will be caused, depending on the GPIRoutSCI bit for the corresponding pad. These bits are sticky bits and are cleared by writing a 1 back to this bit position. The state of GPI_GPE_EN[i] does not prevent the setting of GPI_GPE_STS[i]. Each bit position correspond to 1 pad in the Community: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.1506 Reserved (RSVD7[0])—Offset 138h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1507 Reserved (RSVD7[1])—Offset 13Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1508 Reserved (RSVD7[2])—Offset 140h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1509 Reserved (RSVD7[3])—Offset 144h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1510 Reserved (RSVD7[4])—Offset 148h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1511 Reserved (RSVD7[5])—Offset 14Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1512 GPI General Purpose Events Enable (GPI_GPE_EN_scc_0)—Offset 150h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_206):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
30	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_205):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
29	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_204):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
28	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_203):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
27	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_202):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
26	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_201):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
25	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_200):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
24	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_199):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
23	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_198):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
22	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_197):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
21	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_196):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
20	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_195):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
19	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_194):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
18	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_193):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
17	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_192):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
16	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_191):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
15	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_190):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
14	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_189):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
13	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_210):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
12	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_188):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
11	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_186):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
10	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_185):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_184):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
8	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_183):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
7	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_182):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
6	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_181):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
5	0h RO	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_180):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_179):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
3	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_187):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
2	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_178):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_177):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_176):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.1513 GPI General Purpose Events Enable (GPI_GPE_EN_scc_1)—Offset 154h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RO	**Reserved (RSVD_0):** Reserved
2	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_209):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
1	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_208):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1
0	0h RW	**GPI General Purpose Events Enable (GPI_GPE_EN_GPIO_207):** This bit is used to enable/disable the generation of GPE to cause SCI and/or wake when the corresponding GPI_GPE_STS[i] bit is set. 0 = disable GPE generation 1 = enable GPE generation Note: The pad must also be routed for GPE functionality in order for GPE to be generated, i.e. the corresponding GPIRoutSCI must be set to '1'. Bit assignment: Bit0 = Pad0 Bit1 = Pad1 Bit2 = Pad2 ... Bit N-1= Pad N-1

29.1514 Reserved (RSVD8[0])—Offset 158h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1515 Reserved (RSVD8[1])—Offset 15Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1516 Reserved (RSVD8[2])—Offset 160h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1517 Reserved (RSVD8[3])—Offset 164h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1518 Reserved (RSVD8[4])—Offset 168h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1519 Reserved (RSVD8[5])—Offset 16Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1520 SMI Status (GPI_SMI_STS_scc_0)—Offset 170h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_206):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
30	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_205):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
29	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_204):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
28	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_203):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_202):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
26	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_201):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
25	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_200):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
24	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_199):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
23	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_198):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
22	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_197):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_196):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
20	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_195):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
19	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_194):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_193):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
17	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_192):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
16	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_191):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
15	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_190):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
14	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_189):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
13	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_210):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_188):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
11	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_186):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
10	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_185):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_184):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
8	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_183):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
7	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_182):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_181):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
5	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_180):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
4	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_179):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_187):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
2	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_178):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
1	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_177):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW/1C	**GPI SMI Status (GPI_SMI_STS_GPIO_176):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

29.1521 SMI Status (GPI_SMI_STS_scc_1)—Offset 174h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RO	**Reserved (RSVD_0):** Reserved
2	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_209):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_208):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.
0	0h RO	**GPI SMI Status (GPI_SMI_STS_GPIO_207):** This bit is set to '1' by hardware when a level event (See RxEdCfg, RxInv) is detected, and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding PAD_OWN[2:0] is '000' (i.e. ACPI GPIO Mode). If the following conditions are true, then an SMI will be generated if the GPI_SMI_STS bit is set: 1. The corresponding bit in the GPI_SMI_EN register is set 2. The corresponding pad's GPIROUTSMI is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no SMI event 1 = There is an SMI event The state of GPI_SMI_EN[x] does not prevent the setting of GPI_SMI_STS[x]. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 Defaults for these bits are dependent on the state of the GPI pads.

29.1522 Reserved (RSVD9[0])—Offset 178h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1523 Reserved (RSVD9[1])—Offset 17Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1524 Reserved (RSVD9[2])—Offset 180h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1525 Reserved (RSVD9[3])—Offset 184h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1526 Reserved (RSVD9[4])—Offset 188h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1527 Reserved (RSVD9[5])—Offset 18Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1528 SMI Enable (GPI_SMI_EN_scc_0)—Offset 190h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_206):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
30	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_205):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
29	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_204):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
28	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_203):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
27	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_202):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
26	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_201):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
25	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_200):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
24	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_199):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
23	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_198):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
22	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_197):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_196):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
20	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_195):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
19	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_194):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
18	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_193):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_192):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
16	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_191):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
15	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_190):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
14	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_189):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_210):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
12	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_188):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
11	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_186):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
10	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_185):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
9	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_184):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
8	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_183):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
7	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_182):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
6	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_181):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
5	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_180):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
4	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_179):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
3	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_187):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
2	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_178):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_177):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
0	0h RW	**GPI SMI Enable (GPI_SMI_EN_GPIO_176):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

29.1529 SMI Enable (GPI_SMI_EN_scc_1)—Offset 194h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RO	**Reserved (RSVD_0):** Reserved
2	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_209):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

Bit Range	Default & Access	Field Name (ID): Description
1	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_208):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1
0	0h RO	**GPI SMI Enable (GPI_SMI_EN_GPIO_207):** This bit is used to enable/disable the generation of SMI when the corresponding GPI_SMI_STS bit is set. The pad must also be routed for SMI functionality in order for SMI to be generated, i.e. the corresponding GPIROUTSMI must be set to '1'. 0 = disable SMI generation 1 = enable SMI generation Note: Each of the bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1

29.1530 Reserved (RSVD10[0])—Offset 198h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1531 Reserved (RSVD10[1])—Offset 19Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1532 Reserved (RSVD10[2])—Offset 1A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1533 Reserved (RSVD10[3])—Offset 1A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1534 Reserved (RSVD10[4])—Offset 1A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1535 Reserved (RSVD10[5])—Offset 1ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1536 NMI Status (GPI_NMI_STS_scc_0)—Offset 1B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_206):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
30	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_205):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
29	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_204):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
28	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_203):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
27	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_202):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
26	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_201):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_200):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
24	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_199):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
23	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_198):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
22	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_197):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_196):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
20	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_195):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
19	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_194):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
18	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_193):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
17	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_192):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
16	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_191):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
15	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_190):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
14	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_189):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_210):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
12	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_188):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
11	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_186):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
10	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_185):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_184):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
8	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_183):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
7	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_182):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
6	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_181):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
5	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_180):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
4	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_179):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
3	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_187):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
2	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_178):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_177):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
0	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_176):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.1537 NMI Status (GPI_NMI_STS_scc_1)—Offset 1B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RO	**Reserved (RSVD_0):** Reserved
2	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_209):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
1	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_208):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
0	0h RO	**GPI NMI Status (GPI_NMI_STS_GPIO_207):** This bit is set to '1' by hardware when an edge event is detected (See RxEdCfg, RxInv) on pad and all the following conditions are true: The corresponding pad is used in GPIO input mode (PMode) The corresponding GPIONMIRout is set to '1', i.e. programmed to route as NMI The corresponding GPIOOwn[2:0] is '000' (i.e. ACPI GPIO Mode). The corresponding GPI_NMI_EN is set Writing a value of '1' will clear the bit while writing a value of '0' has no effect. 0 = There is no NMI event 1 = There is an NMI event Bit assignment: Bit0 = pad 0 Bit1 = pad 1 Bit2 = pad 2 ... Bit N-1= pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.1538 Reserved (RSVD11[0])—Offset 1B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1539 Reserved (RSVD11[1])—Offset 1BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1540 Reserved (RSVD11[2])—Offset 1C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1541 Reserved (RSVD11[3])—Offset 1C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1542 Reserved (RSVD11[4])—Offset 1C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1543 Reserved (RSVD11[5])—Offset 1CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**Reserved (RSVD)**

29.1544 NMI Enable (GPI_NMI_EN_scc_0)—Offset 1D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_206):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
30	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_205):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
29	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_204):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
28	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_203):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
27	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_202):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
26	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_201):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
25	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_200):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
24	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_199):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
23	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_198):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
22	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_197):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
21	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_196):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_195):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
19	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_194):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
18	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_193):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
17	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_192):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
16	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_191):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
15	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_190):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
14	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_189):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
13	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_210):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
12	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_188):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
11	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_186):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
10	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_185):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
9	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_184):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
8	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_183):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
7	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_182):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
6	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_181):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
5	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_180):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_179):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
3	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_187):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
2	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_178):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
1	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_177):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_176):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.1545 NMI Enable (GPI_NMI_EN_scc_1)—Offset 1D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RO	**Reserved (RSVD_0):** Reserved
2	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_209):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.
1	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_208):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**GPI NMI Enable (GPI_NMI_EN_GPIO_207):** This bit is used to enable/disable the generation of NMI when the corresponding GPI_NMI_STS bit is set and its GPIROUTNMI is set. 0 = disable NMI generation 1 = enable NMI generation Each bit is lock-able dependent on the associated Pad Configuration register PadCfgLock setting. E.g. if the PadCfgLock of pad0 is '1', bit0 of this bit is locked down to read-only. Bit assignment: Bit0 = Pad 0 Bit1 = Pad 1 Bit2 = Pad 2 ... Bit N-1 = Pad N-1 For pads which do not support NMI, the corresponding bit is read-only zero.

29.1546 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_SMBUS_3P3)—Offset 300h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.1547 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_SMBUS_3P3)—Offset 304h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.1548 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_SMBUS_3P3)—Offset 308h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.1549 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_SMBUS_3P3)—Offset 30Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RW	**strsel B (STRSELB):** Strsel B value
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.1550 Family Configuration Register (FAM_CFG_SMBUS_3P3)—Offset 310h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 5E03000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RW	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
26	1h RW	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
21	1h RW/V	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RW	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	3h RW	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RW	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.1551 Family Reserved Register DW5 (FAM_RSVD_DW5SMBUS_3P3)—Offset 314h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.1552 Family Reserved Register DW6 (FAM_RSVD_DW6SMBUS_3P3)—Offset 318h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.1553 Family Reserved Register DW7 (FAM_RSVD_DW7SMBUS_3P3)—Offset 31Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.1554 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_SDCARD_3P3)—Offset 320h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.1555 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_SDCARD_3P3)—Offset 324h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RO	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.1556 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_SDCARD_3P3)—Offset 328h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.1557 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_SDCARD_3P3)—Offset 32Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RO	**strsel B (STRSELB):** Strsel B value

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.1558 Family Configuration Register (FAM_CFG_SDCARD_3P3)—Offset 330h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 5C00400h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RO	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
26	1h RW	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	0h RO	**strsel (STRSEL):** Pad driver impedance selection
11	0h RW	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8
10	1h RW	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RO	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.1559 Family Reserved Register DW5 (FAM_RSVD_DW5SDCARD_3P3)—Offset 334h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.1560 Family Reserved Register DW6 (FAM_RSVD_DW6SDCARD_3P3)—Offset 338h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.1561 Family Reserved Register DW7 (FAM_RSVD_DW7SDCARD_3P3)—Offset 33Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.1562 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_SD_CLK_1P8)—Offset 340h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.1563 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_SD_CLK_1P8)—Offset 344h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.1564 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_SD_CLK_1P8)—Offset 348h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RO	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.1565 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_SD_CLK_1P8)—Offset 34Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RO	**strsel B (STRSELB):** Strsel B value
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.1566 Family Configuration Register (FAM_CFG_SD_CLK_1P8)—Offset 350h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1C00000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RO	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
26	0h RO	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	0h RW	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8

Bit Range	Default & Access	Field Name (ID): Description
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RO	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.1567 Family Reserved Register DW5 (FAM_RSVD_DW5SD_CLK_1P8)—Offset 354h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.1568 Family Reserved Register DW6 (FAM_RSVD_DW6SD_CLK_1P8)—Offset 358h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.1569 Family Reserved Register DW7 (FAM_RSVD_DW7SD_CLK_1P8)—Offset 35Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.1570 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_CNV_1P8)—Offset 360h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RW	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.1571 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_CNV_1P8)—Offset 364h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.1572 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_CNV_1P8)—Offset 368h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RO	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.1573 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_CNV_1P8)—Offset 36Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RO	**strsel B (STRSELB):** Strsel B value
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.1574 Family Configuration Register (FAM_CFG_CNV_1P8)—Offset 370h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1C03000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RO	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
26	0h RO	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	3h RW	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RO	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.1575 Family Reserved Register DW5 (FAM_RSVD_DW5CNV_1P8)—Offset 374h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.1576 Family Reserved Register DW6 (FAM_RSVD_DW6CNV_1P8)—Offset 378h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.1577 Family Reserved Register DW7 (FAM_RSVD_DW7CNV_1P8)—Offset 37Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.1578 Family RCOMP Register A DW0 (FAM_RCOMP_A_DW0_EMMC_1P8)—Offset 380h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO/V	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RO/V	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.1579 Family RCOMP Register A DW1 (FAM_RCOMP_A_DW1_EMMC_1P8)—Offset 384h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: C0000000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	3h RW	**Static Leg Enable (STATLEGEN):** 0 = Disable 1 = Enable [0] Pull Down (N Leg) [1] Pull Up Leg (P Leg)
29:16	0h RO	**Reserved (RSVD_0):** Reserved
15:13	0h RW	**Reserved (Reserved1):** Reserved
12:8	0h RW	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RW	**Reserved (Reserved2):** Reserved
4:0	0h RW	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.1580 Family RCOMP Register B DW0 (FAM_RCOMP_B_DW0_EMMC_1P8)—Offset 388h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	**Pull Up Termination Strength Value (PUSTRVAL):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:0	0h RO	**Pull Down Termination Strength Value (PDSTRVAL):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.1581 Family RCOMP Register B DW1 (FAM_RCOMP_B_DW1_EMMC_1P8)—Offset 38Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RO	**Reserved (RSVD_0):** Reserved
18:16	0h RO	**strsel B (STRSELB):** Strsel B value

Bit Range	Default & Access	Field Name (ID): Description
15:13	0h RO	**Reserved (Reserved1):** Reserved
12:8	0h RO	**Pull Up Slew Rate Value (PUSLEWVAL):** [12:8] Pull Up Slew Rate value
7:5	0h RO	**Reserved (Reserved2):** Reserved
4:0	0h RO	**Pull Down Slew Rate Value (PDSLEWVAL):** [4:0] Pull Down Slew Rate Value

29.1582 Family Configuration Register (FAM_CFG_EMMC_1P8)—Offset 390h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1C02000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**drvmode (DRVMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
29:28	0h RO	**i2cen (I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
27	0h RO	**csen (CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
26	0h RO	**parkmodeen_b (PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
25:24	1h RW	**hysctl (HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
23	1h RO	**AZAMODELVHB (AZAMODELVHB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
22	1h RO	**clinkenb (CLINKENB):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
21	0h RO	**v1p8mode (V1P8MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**v1p5mode (V1P5MODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**hsmode (HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**odtupdn (ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
17	0h RO	**odten (ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
16	0h RO	**analogmuxen (ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the Pad Register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are family specific. Bits that aren't implemented or output of the CFIO are read only.
15	0h RO	**padtol (PADTOL):** PAD voltage tolerance: When padtol=0, v1p8mode = PAD and VCCIO both .3Vtolerance. v1p8mode = PAD and VCCIO both .8Vtolerance. When padtol=1, v1p8mode = PAD 1.8V tolerance and VCCIO .3Vtolerance.
14:12	2h RW	**strsel (STRSEL):** Pad driver impedance selection
11	0h RO	**Weak Leaker Pull-up (WEAKLEVEL):** v1p8mode = 0, Weak leaker Pull up to 0: vcca3p3_uhv 1: (vcca3p3_uhv - vtp). v1p8mode = 1, Weak pullup is always vcca1p8
10	0h RO	**floating ESD (FLOATINGESD):** If set to 1, the ESD supply will float between 3.3 and 1.8v to reduce power and provide isolation during ESD events. If set to 0, the ESD supply is shorted to 3.3v. During DFx mode like BSCAN, SCAN and etc, the ESD control shall be constrained to 0
9:3	0h RO	**Reserved (RSVD_0):** Reserved
2:0	0h RO	**Current Source Strength (CURRSRCSTR):** Current Source Strength configuration bits for I2C High Speed Mode. Only supported on ULPMSMV CFIO Type. Connected to crtstr[2:0] CFIO signal.

29.1583 Family Reserved Register DW5 (FAM_RSVD_DW5EMMC_1P8)—Offset 394h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW5

29.1584 Family Reserved Register DW6 (FAM_RSVD_DW6EMMC_1P8)—Offset 398h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW6

29.1585 Family Reserved Register DW7 (FAM_RSVD_DW7EMMC_1P8)—Offset 39Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** FAM reserved register DW7

29.1586 RCOMP Control Register (RCOMP_CTL_SMBUS_3P3)—Offset 480h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 94087903h

Bit Range	Default & Access	Field Name (ID): Description
31	1h RW	**RCOMP FSM Enable (RFSMEN):** 0 = state machine disabled (clock gated) 1 = state machine enabled It is important to have this bit enabled in order for all internal clocks inside the RCOMP FSM to run.
30	0h RW	**RCOMP Calibration Start (RCALSTART):** When set to 1 this bit triggers the RCOMP FSM to start hunting for the correct values for strength and slew. This bit should be cleared after calibration has finished. Before setting the RcalStart bit RFSMEn must have been asserted for at least 5 clock cycles. The first calibration takes 10us to complete and subsequent calibrations take 3us. Details of this process are in the CFIP RCOMP Aspec.
29:28	1h RW	**RCOMP FSM Stop Condition (RFSMSTOP):** RCOMP FSM Stop calibration condition. RCOMP FSM (pullup & pulldown) will finish calibrating once orcpupen/orcpdnen toggles. The toggling indicates the compensation value has been reached. This option is programmable for toggle once (single edge), toggle twice (double edge), toggle thrice (triple edge) or forcing it to finish after a certain cycle (defined by RFSMStopCyc). 00b = single edge detect (10 or 01) 01b = double edge detect (101 or 010) (default) 10b = triple edge detect (1010 or 0101) 11b = stop after x cycles (depending on ircstpcyc[3:0] value -- This is meant for debug)
27:24	4h RW	**RCOMP FSM Stop Cycles (RFSMSTOPCYC):** Number of cycles to stop RCOMP FSM from searching for strength and slew settings. RFSMStop must be set to 11b for this option to take effect. 0000b = 0 cycle (illegal option) 0001b = 1 cycle (illegal option) 0010b = 2 cycles (illegal option) 0011b = 3 cycles (illegal option) 0100b = 4 cycles 1110b = 14 cycles 1111b = 15 cycles (maximum allowed) This is meant for debug.
23:20	0h RO	**Reserved (RSVD_0):** Reserved
19:11	10Fh RW	**Initial Pull Up Calibration Value (INITPUCALVAL):** Initial pleg and nleg calibration values. RCOMP FSM will always start with these values for pullup and pulldown calibration. [8:0] Initial Pull Down Value (nleg) [10:9] Reserved [19:11] Initial Pull Up Value (pleg) [21:20] Reserved
10:9	0h RO	**Reserved (RSVD_1):** Reserved
8:0	103h RW	**Initial Pull Down Calibration Value (INITPDCALVAL):** Initial pleg and nleg calibration values. RCOMP FSM will always start with these values for pullup and pulldown calibration. [8:0] Initial Pull Down Value (nleg) [10:9] Reserved [19:11] Initial Pull Up Value (pleg) [21:20] Reserved

29.1587 RCOMP Offset Register (RCOMP_OFFSET_SMBUS_3P3)—Offset 484h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:24	0h RW	**Pull Up Offset Slew Value (PUOFFSLEWVAL):** Offset strength value for pleg and nleg. [5:0] Pull Down Slew Offset (nleg). Bit 4 is sign bit. [7:5] Reserved [13:8] Pull Up Slew Offset (pleg). Bit 12 is sign bit. [15:13] Reserved
23:22	0h RO	**Reserved (RSVD_1):** Reserved
21:16	0h RW	**Pull Down Offset Slew Value (PDOFFSLEWVAL):** Offset strength value for pleg and nleg. [5:0] Pull Down Slew Offset (nleg). Bit 4 is sign bit. [7:5] Reserved [13:8] Pull Up Slew Offset (pleg). Bit 12 is sign bit. [15:13] Reserved
15:14	0h RO	**Reserved (RSVD_2):** Reserved
13:8	0h RW	**Pull Up Offset Strength Value (PUOFFSTRVAL):** Offset strength value for pleg and nleg. [5:0] Pull Down Strength Offset (nleg). Bit 5 is sign bit. [7:6] Reserved [13:8] Pull Up Strength Offset (pleg). Bit 13 is sign bit. [15:14]Reserved
7:6	0h RO	**Reserved (RSVD_3):** Reserved
5:0	0h RW	**Pull Down Offset Strength Value (PDOFFSTRVAL):** Offset strength value for pleg and nleg. [5:0] Pull Down Strength Offset (nleg). Bit 5 is sign bit. [7:6] Reserved [13:8] Pull Up Strength Offset (pleg). Bit 13 is sign bit. [15:14]Reserved

29.1588 RCOMP Override Registers DW0 (RCOMP_OVR_DW0_SMBUS_3P3)—Offset 488h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 410041h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD_0):** Reserved
24:16	41h RW	**Pull Up Override Strength Value (OVRSTRPUVAL):** Override strength value for pullup and pulldown settings. Should be 50ohm typical corner. [8:0] Pull Down Strength Value (nleg) [10:9] Reserved [19:11] Pull Up Strength Value (pleg) [21:20] Reserved
15:9	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
8:0	41h RW	**Pull Down Override Strength Value (OVRSTRPDVAL):** Override strength value for pullup and pulldown settings. Should be 50ohm typical corner. [8:0] Pull Down Strength Value (nleg) [10:9] Reserved [19:11] Pull Up Strength Value (pleg) [21:20] Reserved

29.1589 RCOMP Override Registers DW1 (RCOMP_OVR_DW1_SMBUS_3P3)—Offset 48Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Override RCOMP Enable (OVRREN):** Override enable bit to override the strength and slew values driven by the RCOMP FSM. The override values are driven through config registers. 0 = Override Disable 1 = Override Enable
30	0h RW	**Override Load (OVRLOAD):** Load the override values into the RCOMP FSM and drive these values to the CFIO buffer. This is only valid when OvrRen = 1. 0 = Unload Values 1 = Load Values to CFIO
29:13	0h RO	**Reserved (RSVD_0):** Reserved
12:8	0h RW	**Pull Up Override Slew Value (PUOVRSLEWVAL):** Override slew values for pullup and pulldown settings. [4:0] Pull Down Slew Value (nleg) [12:8] Pull Up Slew Value (pleg)
7:5	0h RO	**Reserved (RSVD_1):** Reserved
4:0	0h RW	**Pull Down Override Slew Value (PDOVRSLEWVAL):** Override slew values for pullup and pulldown settings. [4:0] Pull Down Slew Value (nleg) [12:8] Pull Up Slew Value (pleg)

29.1590 RCOMP Value Register DW0 (RCOMP_VALUE_DW0_SMBUS_3P3)—Offset 490h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
24:16	0h RO	**Pull Up Strength Value (STRVALPLEG):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:9	0h RO	**Reserved (RSVD_1):** Reserved
8:0	0h RO	**Pull Down Strength Value (STRVALNLEG):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.1591 RCOMP Value Register DW1 (RCOMP_VALUE_DW1_SMBUS_3P3)—Offset 494h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:13	0h RO	**Reserved (RSVD_0):** Reserved
12:8	0h RO	**Pull Up Slew Value (RCOMP_PUSLEWVAL):** Slew values. Valid only after calibration is done. [4:0] Pull Down Slew Value (nleg) [12:8] Pull Up Slew Value (pleg)
7:5	0h RO	**Reserved (RSVD_1):** Reserved
4:0	0h RO	**Pull Down Slew Value (RCOMP_PDSLEWVAL):** Slew values. Valid only after calibration is done. [4:0] Pull Down Slew Value (nleg) [12:8] Pull Up Slew Value (pleg)

29.1592 RCOMP Control Register DW1 (RCOMP_CTL_DW1_SMBUS_3P3)—Offset 498h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 50277h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD_0):** Reserved
24	0h RO/V	**RCOMP Calibration Ongoing (RCALON):** 0 = RCOMP FSM is idle 1 = RCOMP calibration is ongoing
23:19	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
18:16	5h RW	**RCOMP Oversampling Count (RCOMP_ROVERSAMPCNT):** 000 Illegal 001 3x DDR sampling 010 5x DDR sampling 011 7x DDR sampling 100 9x DDR sampling 101 11x DDR sampling 110 13x DDR sampling 111 15x DDR sampling
15:13	0h RO	**Reserved (RSVD_2):** Reserved
12:8	2h RW	**RCOMP Aging Activity Factor (RCOMP_RAGEAF):** 00000 -- 0% 00001 -- 5% 00010 -- 10% 00011 -- 15% 00100 -- 20% 00101 -- 25% 00110 -- 30% 00111 -- 35% 01000 -- 40% 01001 -- 45% 01010 -- 50%
7:4	7h RW	**RCOMP Interval Select (RINTSEL):** This field specifies the interval for RCOMP calibration. 0000 - non-periodic mode 0.5ms - 0001 (illegal option) 1ms - 0010 2ms - 0011 3ms - 0100 4ms - 0101 5ms - 0110 10ms - 0111 15ms - 1000 25ms - 1001 50ms - 1010 100ms - 1011 200ms - 1100 500ms - 1101 1000ms - 1110 2000ms - 1111
3	0h RO	**Reserved (RSVD_3):** Reserved
2	1h RW	**halfbit feature enable. (HALFBITEN):** 1: enable halfbit feature 0: disable halfbit feature
1	1h RW	**RCOMP A0 FSM and E0 FSM selection (RCOMP_RFSMSEL):** 1 Choose E0 FSM 0 Choose A0 FSM
0	1h RW	**RCOMP Global Reset (RFSMRST):** RCOMP FSM reset signal, active low.

29.1593 RCOMP Reserved Register (RCOMP_RSVD_SMBUS_3P3)—Offset 49Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** RCOMP reserved register

29.1594 RCOMP Control Register (RCOMP_CTL_SDCARD_3P3)—Offset 4A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 94087903h

Bit Range	Default & Access	Field Name (ID): Description
31	1h RW	**RCOMP FSM Enable (RFSMEN):** 0 = state machine disabled (clock gated) 1 = state machine enabled It is important to have this bit enabled in order for all internal clocks inside the RCOMP FSM to run.
30	0h RW	**RCOMP Calibration Start (RCALSTART):** When set to 1 this bit triggers the RCOMP FSM to start hunting for the correct values for strength and slew. This bit should be cleared after calibration has finished. Before setting the RcalStart bit RFSMEn must have been asserted for at least 5 clock cycles. The first calibration takes 10us to complete and subsequent calibrations take 3us. Details of this process are in the CFIP RCOMP Aspec.
29:28	1h RW	**RCOMP FSM Stop Condition (RFSMSTOP):** RCOMP FSM Stop calibration condition. RCOMP FSM (pullup & pulldown) will finish calibrating once orcpupen/orcpdnen toggles. The toggling indicates the compensation value has been reached. This option is programmable for toggle once (single edge), toggle twice (double edge), toggle thrice (triple edge) or forcing it to finish after a certain cycle (defined by RFSMStopCyc). 00b = single edge detect (10 or 01) 01b = double edge detect (101 or 010) (default) 10b = triple edge detect (1010 or 0101) 11b = stop after x cycles (depending on ircstpcyc[3:0] value -- This is meant for debug)
27:24	4h RW	**RCOMP FSM Stop Cycles (RFSMSTOPCYC):** Number of cycles to stop RCOMP FSM from searching for strength and slew settings. RFSMStop must be set to 11b for this option to take effect. 0000b = 0 cycle (illegal option) 0001b = 1 cycle (illegal option) 0010b = 2 cycles (illegal option) 0011b = 3 cycles (illegal option) 0100b = 4 cycles 1110b = 14 cycles 1111b = 15 cycles (maximum allowed) This is meant for debug.
23:20	0h RO	**Reserved (RSVD_0):** Reserved
19:11	10Fh RW	**Initial Pull Up Calibration Value (INITPUCALVAL):** Initial pleg and nleg calibration values. RCOMP FSM will always start with these values for pullup and pulldown calibration. [8:0] Initial Pull Down Value (nleg) [10:9] Reserved [19:11] Initial Pull Up Value (pleg) [21:20] Reserved
10:9	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
8:0	103h RW	**Initial Pull Down Calibration Value (INITPDCALVAL):** Initial pleg and nleg calibration values. RCOMP FSM will always start with these values for pullup and pulldown calibration. [8:0] Initial Pull Down Value (nleg) [10:9] Reserved [19:11] Initial Pull Up Value (pleg) [21:20] Reserved

29.1595 RCOMP Offset Register (RCOMP_OFFSET_SDCARD_3P3)—Offset 4A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:24	0h RW	**Pull Up Offset Slew Value (PUOFFSLEWVAL):** Offset strength value for pleg and nleg. [5:0] Pull Down Slew Offset (nleg). Bit 4 is sign bit. [7:5] Reserved [13:8] Pull Up Slew Offset (pleg). Bit 12 is sign bit. [15:13] Reserved
23:22	0h RO	**Reserved (RSVD_1):** Reserved
21:16	0h RW	**Pull Down Offset Slew Value (PDOFFSLEWVAL):** Offset strength value for pleg and nleg. [5:0] Pull Down Slew Offset (nleg). Bit 4 is sign bit. [7:5] Reserved [13:8] Pull Up Slew Offset (pleg). Bit 12 is sign bit. [15:13] Reserved
15:14	0h RO	**Reserved (RSVD_2):** Reserved
13:8	0h RW	**Pull Up Offset Strength Value (PUOFFSTRVAL):** Offset strength value for pleg and nleg. [5:0] Pull Down Strength Offset (nleg). Bit 5 is sign bit. [7:6] Reserved [13:8] Pull Up Strength Offset (pleg). Bit 13 is sign bit. [15:14]Reserved
7:6	0h RO	**Reserved (RSVD_3):** Reserved
5:0	0h RW	**Pull Down Offset Strength Value (PDOFFSTRVAL):** Offset strength value for pleg and nleg. [5:0] Pull Down Strength Offset (nleg). Bit 5 is sign bit. [7:6] Reserved [13:8] Pull Up Strength Offset (pleg). Bit 13 is sign bit. [15:14]Reserved

29.1596 RCOMP Override Registers DW0 (RCOMP_OVR_DW0_SDCARD_3P3)—Offset 4A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 410041h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD_0):** Reserved
24:16	41h RW	**Pull Up Override Strength Value (OVRSTRPUVAL):** Override strength value for pullup and pulldown settings. Should be 50ohm typical corner. [8:0] Pull Down Strength Value (nleg) [10:9] Reserved [19:11] Pull Up Strength Value (pleg) [21:20] Reserved
15:9	0h RO	**Reserved (RSVD_1):** Reserved
8:0	41h RW	**Pull Down Override Strength Value (OVRSTRPDVAL):** Override strength value for pullup and pulldown settings. Should be 50ohm typical corner. [8:0] Pull Down Strength Value (nleg) [10:9] Reserved [19:11] Pull Up Strength Value (pleg) [21:20] Reserved

29.1597 RCOMP Override Registers DW1 (RCOMP_OVR_DW1_SDCARD_3P3)—Offset 4ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Override RCOMP Enable (OVRREN):** Override enable bit to override the strength and slew values driven by the RCOMP FSM. The override values are driven through config registers. 0 = Override Disable 1 = Override Enable
30	0h RW	**Override Load (OVRLOAD):** Load the override values into the RCOMP FSM and drive these values to the CFIO buffer. This is only valid when OvrRen = 1. 0 = Unload Values 1 = Load Values to CFIO
29:13	0h RO	**Reserved (RSVD_0):** Reserved
12:8	0h RW	**Pull Up Override Slew Value (PUOVRSLEWVAL):** Override slew values for pullup and pulldown settings. [4:0] Pull Down Slew Value (nleg) [12:8] Pull Up Slew Value (pleg)
7:5	0h RO	**Reserved (RSVD_1):** Reserved
4:0	0h RW	**Pull Down Override Slew Value (PDOVRSLEWVAL):** Override slew values for pullup and pulldown settings. [4:0] Pull Down Slew Value (nleg) [12:8] Pull Up Slew Value (pleg)

29.1598 RCOMP Value Register DW0 (RCOMP_VALUE_DW0_SDCARD_3P3)—Offset 4B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD_0):** Reserved
24:16	0h RO	**Pull Up Strength Value (STRVALPLEG):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:9	0h RO	**Reserved (RSVD_1):** Reserved
8:0	0h RO	**Pull Down Strength Value (STRVALNLEG):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.1599 RCOMP Value Register DW1 (RCOMP_VALUE_DW1_SDCARD_3P3)—Offset 4B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:13	0h RO	**Reserved (RSVD_0):** Reserved
12:8	0h RO	**Pull Up Slew Value (RCOMP_PUSLEWVAL):** Slew values. Valid only after calibration is done. [4:0] Pull Down Slew Value (nleg) [12:8] Pull Up Slew Value (pleg)
7:5	0h RO	**Reserved (RSVD_1):** Reserved
4:0	0h RO	**Pull Down Slew Value (RCOMP_PDSLEWVAL):** Slew values. Valid only after calibration is done. [4:0] Pull Down Slew Value (nleg) [12:8] Pull Up Slew Value (pleg)

29.1600 RCOMP Control Register DW1 (RCOMP_CTL_DW1_SDCARD_3P3)—Offset 4B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 50273h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD_0):** Reserved
24	0h RO/V	**RCOMP Calibration Ongoing (RCALON):** 0 = RCOMP FSM is idle 1 = RCOMP calibration is ongoing
23:19	0h RO	**Reserved (RSVD_1):** Reserved
18:16	5h RW	**RCOMP Oversampling Count (RCOMP_ROVERSAMPCNT):** 000 Illegal 001 3x DDR sampling 010 5x DDR sampling 011 7x DDR sampling 100 9x DDR sampling 101 11x DDR sampling 110 13x DDR sampling 111 15x DDR sampling
15:13	0h RO	**Reserved (RSVD_2):** Reserved
12:8	2h RW	**RCOMP Aging Activity Factor (RCOMP_RAGEAF):** 00000 -- 0% 00001 -- 5% 00010 -- 10% 00011 -- 15% 00100 -- 20% 00101 -- 25% 00110 -- 30% 00111 -- 35% 01000 -- 40% 01001 -- 45% 01010 -- 50%
7:4	7h RW	**RCOMP Interval Select (RINTSEL):** This field specifies the interval for RCOMP calibration. 0000 - non-periodic mode 0.5ms - 0001 (illegal option) 1ms - 0010 2ms - 0011 3ms - 0100 4ms - 0101 5ms - 0110 10ms - 0111 15ms - 1000 25ms - 1001 50ms - 1010 100ms - 1011 200ms - 1100 500ms - 1101 1000ms - 1110 2000ms - 1111
3	0h RO	**Reserved (RSVD_3):** Reserved
2	0h RW	**halfbit feature enable. (HALFBITEN):** 1: enable halfbit feature 0: disable halfbit feature
1	1h RW	**RCOMP A0 FSM and E0 FSM selection (RCOMP_RFSMSEL):** 1 Choose E0 FSM 0 Choose A0 FSM
0	1h RW	**RCOMP Global Reset (RFSMRST):** RCOMP FSM reset signal, active low.

29.1601 RCOMP Reserved Register (RCOMP_RSVD_SDCARD_3P3)—Offset 4BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** RCOMP reserved register

29.1602 RCOMP Control Register (RCOMP_CTL_SD_CLK_1P8)—Offset 4C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 94087903h

Bit Range	Default & Access	Field Name (ID): Description
31	1h RW	**RCOMP FSM Enable (RFSMEN):** 0 = state machine disabled (clock gated) 1 = state machine enabled It is important to have this bit enabled in order for all internal clocks inside the RCOMP FSM to run.
30	0h RW	**RCOMP Calibration Start (RCALSTART):** When set to 1 this bit triggers the RCOMP FSM to start hunting for the correct values for strength and slew. This bit should be cleared after calibration has finished. Before setting the RcalStart bit RFSMEn must have been asserted for at least 5 clock cycles. The first calibration takes 10us to complete and subsequent calibrations take 3us. Details of this process are in the CFIP RCOMP Aspec.
29:28	1h RW	**RCOMP FSM Stop Condition (RFSMSTOP):** RCOMP FSM Stop calibration condition. RCOMP FSM (pullup & pulldown) will finish calibrating once orcpupen/orcpdnen toggles. The toggling indicates the compensation value has been reached. This option is programmable for toggle once (single edge), toggle twice (double edge), toggle thrice (triple edge) or forcing it to finish after a certain cycle (defined by RFSMStopCyc). 00b = single edge detect (10 or 01) 01b = double edge detect (101 or 010) (default) 10b = triple edge detect (1010 or 0101) 11b = stop after x cycles (depending on ircstpcyc[3:0] value -- This is meant for debug)
27:24	4h RW	**RCOMP FSM Stop Cycles (RFSMSTOPCYC):** Number of cycles to stop RCOMP FSM from searching for strength and slew settings. RFSMStop must be set to 11b for this option to take effect. 0000b = 0 cycle (illegal option) 0001b = 1 cycle (illegal option) 0010b = 2 cycles (illegal option) 0011b = 3 cycles (illegal option) 0100b = 4 cycles 1110b = 14 cycles 1111b = 15 cycles (maximum allowed) This is meant for debug.
23:20	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
19:11	10Fh RW	**Initial Pull Up Calibration Value (INITPUCALVAL):** Initial pleg and nleg calibration values. RCOMP FSM will always start with these values for pullup and pulldown calibration. [8:0] Initial Pull Down Value (nleg) [10:9] Reserved [19:11] Initial Pull Up Value (pleg) [21:20] Reserved
10:9	0h RO	**Reserved (RSVD_1):** Reserved
8:0	103h RW	**Initial Pull Down Calibration Value (INITPDCALVAL):** Initial pleg and nleg calibration values. RCOMP FSM will always start with these values for pullup and pulldown calibration. [8:0] Initial Pull Down Value (nleg) [10:9] Reserved [19:11] Initial Pull Up Value (pleg) [21:20] Reserved

29.1603 RCOMP Offset Register (RCOMP_OFFSET_SD_CLK_1P8)—Offset 4C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:24	0h RW	**Pull Up Offset Slew Value (PUOFFSLEWVAL):** Offset strength value for pleg and nleg. [5:0] Pull Down Slew Offset (nleg). Bit 4 is sign bit. [7:5] Reserved [13:8] Pull Up Slew Offset (pleg). Bit 12 is sign bit. [15:13] Reserved
23:22	0h RO	**Reserved (RSVD_1):** Reserved
21:16	0h RW	**Pull Down Offset Slew Value (PDOFFSLEWVAL):** Offset strength value for pleg and nleg. [5:0] Pull Down Slew Offset (nleg). Bit 4 is sign bit. [7:5] Reserved [13:8] Pull Up Slew Offset (pleg). Bit 12 is sign bit. [15:13] Reserved
15:14	0h RO	**Reserved (RSVD_2):** Reserved
13:8	0h RW	**Pull Up Offset Strength Value (PUOFFSTRVAL):** Offset strength value for pleg and nleg. [5:0] Pull Down Strength Offset (nleg). Bit 5 is sign bit. [7:6] Reserved [13:8] Pull Up Strength Offset (pleg). Bit 13 is sign bit. [15:14]Reserved
7:6	0h RO	**Reserved (RSVD_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
5:0	0h RW	**Pull Down Offset Strength Value (PDOFFSTRVAL):** Offset strength value for pleg and nleg. [5:0] Pull Down Strength Offset (nleg). Bit 5 is sign bit. [7:6] Reserved [13:8] Pull Up Strength Offset (pleg). Bit 13 is sign bit. [15:14]Reserved

29.1604 RCOMP Override Registers DW0 (RCOMP_OVR_DW0_SD_CLK_1P8)—Offset 4C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 410041h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD_0):** Reserved
24:16	41h RW	**Pull Up Override Strength Value (OVRSTRPUVAL):** Override strength value for pullup and pulldown settings. Should be 50ohm typical corner. [8:0] Pull Down Strength Value (nleg) [10:9] Reserved [19:11] Pull Up Strength Value (pleg) [21:20] Reserved
15:9	0h RO	**Reserved (RSVD_1):** Reserved
8:0	41h RW	**Pull Down Override Strength Value (OVRSTRPDVAL):** Override strength value for pullup and pulldown settings. Should be 50ohm typical corner. [8:0] Pull Down Strength Value (nleg) [10:9] Reserved [19:11] Pull Up Strength Value (pleg) [21:20] Reserved

29.1605 RCOMP Override Registers DW1 (RCOMP_OVR_DW1_SD_CLK_1P8)—Offset 4CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Override RCOMP Enable (OVRREN):** Override enable bit to override the strength and slew values driven by the RCOMP FSM. The override values are driven through config registers. 0 = Override Disable 1 = Override Enable
30	0h RW	**Override Load (OVRLOAD):** Load the override values into the RCOMP FSM and drive these values to the CFIO buffer. This is only valid when OvrRen = 1. 0 = Unload Values 1 = Load Values to CFIO

Bit Range	Default & Access	Field Name (ID): Description
29:13	0h RO	**Reserved (RSVD_0):** Reserved
12:8	0h RW	**Pull Up Override Slew Value (PUOVRSLEWVAL):** Override slew values for pullup and pulldown settings. [4:0] Pull Down Slew Value (nleg) [12:8] Pull Up Slew Value (pleg)
7:5	0h RO	**Reserved (RSVD_1):** Reserved
4:0	0h RW	**Pull Down Override Slew Value (PDOVRSLEWVAL):** Override slew values for pullup and pulldown settings. [4:0] Pull Down Slew Value (nleg) [12:8] Pull Up Slew Value (pleg)

29.1606 RCOMP Value Register DW0 (RCOMP_VALUE_DW0_SD_CLK_1P8)—Offset 4D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD_0):** Reserved
24:16	0h RO	**Pull Up Strength Value (STRVALPLEG):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:9	0h RO	**Reserved (RSVD_1):** Reserved
8:0	0h RO	**Pull Down Strength Value (STRVALNLEG):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.1607 RCOMP Value Register DW1 (RCOMP_VALUE_DW1_SD_CLK_1P8)—Offset 4D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:13	0h RO	**Reserved (RSVD_0):** Reserved
12:8	0h RO	**Pull Up Slew Value (RCOMP_PUSLEWVAL):** Slew values. Valid only after calibration is done. [4:0] Pull Down Slew Value (nleg) [12:8] Pull Up Slew Value (pleg)

Bit Range	Default & Access	Field Name (ID): Description
7:5	0h RO	**Reserved (RSVD_1):** Reserved
4:0	0h RO	**Pull Down Slew Value (RCOMP_PDSLEWVAL):** Slew values. Valid only after calibration is done. [4:0] Pull Down Slew Value (nleg) [12:8] Pull Up Slew Value (pleg)

29.1608 RCOMP Control Register DW1 (RCOMP_CTL_DW1_SD_CLK_1P8)—Offset 4D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 50277h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD_0):** Reserved
24	0h RO/V	**RCOMP Calibration Ongoing (RCALON):** 0 = RCOMP FSM is idle 1 = RCOMP calibration is ongoing
23:19	0h RO	**Reserved (RSVD_1):** Reserved
18:16	5h RW	**RCOMP Oversampling Count (RCOMP_ROVERSAMPCNT):** 000 Illegal 001 3x DDR sampling 010 5x DDR sampling 011 7x DDR sampling 100 9x DDR sampling 101 11x DDR sampling 110 13x DDR sampling 111 15x DDR sampling
15:13	0h RO	**Reserved (RSVD_2):** Reserved
12:8	2h RW	**RCOMP Aging Activity Factor (RCOMP_RAGEAF):** 00000 -- 0% 00001 -- 5% 00010 -- 10% 00011 -- 15% 00100 -- 20% 00101 -- 25% 00110 -- 30% 00111 -- 35% 01000 -- 40% 01001 -- 45% 01010 -- 50%

Bit Range	Default & Access	Field Name (ID): Description
7:4	7h RW	**RCOMP Interval Select (RINTSEL):** This field specifies the interval for RCOMP calibration. 0000 - non-periodic mode 0.5ms - 0001 (illegal option) 1ms - 0010 2ms - 0011 3ms - 0100 4ms - 0101 5ms - 0110 10ms - 0111 15ms - 1000 25ms - 1001 50ms - 1010 100ms - 1011 200ms - 1100 500ms - 1101 1000ms - 1110 2000ms - 1111
3	0h RO	**Reserved (RSVD_3):** Reserved
2	1h RW	**halfbit feature enable. (HALFBITEN):** 1: enable halfbit feature 0: disable halfbit feature
1	1h RW	**RCOMP A0 FSM and E0 FSM selection (RCOMP_RFSMSEL):** 1 Choose E0 FSM 0 Choose A0 FSM
0	1h RW	**RCOMP Global Reset (RFSMRST):** RCOMP FSM reset signal, active low.

29.1609 RCOMP Reserved Register (RCOMP_RSVD_SD_CLK_1P8)—Offset 4DCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** RCOMP reserved register

29.1610 RCOMP Control Register (RCOMP_CTL_EMMC_1P8)—Offset 4E0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 94087903h

Bit Range	Default & Access	Field Name (ID): Description
31	1h RW	**RCOMP FSM Enable (RFSMEN):** 0 = state machine disabled (clock gated) 1 = state machine enabled It is important to have this bit enabled in order for all internal clocks inside the RCOMP FSM to run.
30	0h RW	**RCOMP Calibration Start (RCALSTART):** When set to 1 this bit triggers the RCOMP FSM to start hunting for the correct values for strength and slew. This bit should be cleared after calibration has finished. Before setting the RcalStart bit RFSMEn must have been asserted for at least 5 clock cycles. The first calibration takes 10us to complete and subsequent calibrations take 3us. Details of this process are in the CFIP RCOMP Aspec.
29:28	1h RW	**RCOMP FSM Stop Condition (RFSMSTOP):** RCOMP FSM Stop calibration condition. RCOMP FSM (pullup & pulldown) will finish calibrating once orcpupen/orcpdnen toggles. The toggling indicates the compensation value has been reached. This option is programmable for toggle once (single edge), toggle twice (double edge), toggle thrice (triple edge) or forcing it to finish after a certain cycle (defined by RFSMStopCyc). 00b = single edge detect (10 or 01) 01b = double edge detect (101 or 010) (default) 10b = triple edge detect (1010 or 0101) 11b = stop after x cycles (depending on ircstpcyc[3:0] value -- This is meant for debug)
27:24	4h RW	**RCOMP FSM Stop Cycles (RFSMSTOPCYC):** Number of cycles to stop RCOMP FSM from searching for strength and slew settings. RFSMStop must be set to 11b for this option to take effect. 0000b = 0 cycle (illegal option) 0001b = 1 cycle (illegal option) 0010b = 2 cycles (illegal option) 0011b = 3 cycles (illegal option) 0100b = 4 cycles 1110b = 14 cycles 1111b = 15 cycles (maximum allowed) This is meant for debug.
23:20	0h RO	**Reserved (RSVD_0):** Reserved
19:11	10Fh RW	**Initial Pull Up Calibration Value (INITPUCALVAL):** Initial pleg and nleg calibration values. RCOMP FSM will always start with these values for pullup and pulldown calibration. [8:0] Initial Pull Down Value (nleg) [10:9] Reserved [19:11] Initial Pull Up Value (pleg) [21:20] Reserved
10:9	0h RO	**Reserved (RSVD_1):** Reserved
8:0	103h RW	**Initial Pull Down Calibration Value (INITPDCALVAL):** Initial pleg and nleg calibration values. RCOMP FSM will always start with these values for pullup and pulldown calibration. [8:0] Initial Pull Down Value (nleg) [10:9] Reserved [19:11] Initial Pull Up Value (pleg) [21:20] Reserved

29.1611 RCOMP Offset Register (RCOMP_OFFSET_EMMC_1P8)—Offset 4E4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**Reserved (RSVD_0):** Reserved
29:24	0h RW	**Pull Up Offset Slew Value (PUOFFSLEWVAL):** Offset strength value for pleg and nleg. [5:0] Pull Down Slew Offset (nleg). Bit 4 is sign bit. [7:5] Reserved [13:8] Pull Up Slew Offset (pleg). Bit 12 is sign bit. [15:13] Reserved
23:22	0h RO	**Reserved (RSVD_1):** Reserved
21:16	0h RW	**Pull Down Offset Slew Value (PDOFFSLEWVAL):** Offset strength value for pleg and nleg. [5:0] Pull Down Slew Offset (nleg). Bit 4 is sign bit. [7:5] Reserved [13:8] Pull Up Slew Offset (pleg). Bit 12 is sign bit. [15:13] Reserved
15:14	0h RO	**Reserved (RSVD_2):** Reserved
13:8	0h RW	**Pull Up Offset Strength Value (PUOFFSTRVAL):** Offset strength value for pleg and nleg. [5:0] Pull Down Strength Offset (nleg). Bit 5 is sign bit. [7:6] Reserved [13:8] Pull Up Strength Offset (pleg). Bit 13 is sign bit. [15:14]Reserved
7:6	0h RO	**Reserved (RSVD_3):** Reserved
5:0	0h RW	**Pull Down Offset Strength Value (PDOFFSTRVAL):** Offset strength value for pleg and nleg. [5:0] Pull Down Strength Offset (nleg). Bit 5 is sign bit. [7:6] Reserved [13:8] Pull Up Strength Offset (pleg). Bit 13 is sign bit. [15:14]Reserved

29.1612 RCOMP Override Registers DW0 (RCOMP_OVR_DW0_EMMC_1P8)—Offset 4E8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 410041h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD_0):** Reserved
24:16	41h RW	**Pull Up Override Strength Value (OVRSTRPUVAL):** Override strength value for pullup and pulldown settings. Should be 50ohm typical corner. [8:0] Pull Down Strength Value (nleg) [10:9] Reserved [19:11] Pull Up Strength Value (pleg) [21:20] Reserved
15:9	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
8:0	41h RW	**Pull Down Override Strength Value (OVRSTRPDVAL):** Override strength value for pullup and pulldown settings. Should be 50ohm typical corner. [8:0] Pull Down Strength Value (nleg) [10:9] Reserved [19:11] Pull Up Strength Value (pleg) [21:20] Reserved

29.1613 RCOMP Override Registers DW1 (RCOMP_OVR_DW1_EMMC_1P8)—Offset 4ECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Override RCOMP Enable (OVRREN):** Override enable bit to override the strength and slew values driven by the RCOMP FSM. The override values are driven through config registers. 0 = Override Disable 1 = Override Enable
30	0h RW	**Override Load (OVRLOAD):** Load the override values into the RCOMP FSM and drive these values to the CFIO buffer. This is only valid when OvrRen = 1. 0 = Unload Values 1 = Load Values to CFIO
29:13	0h RO	**Reserved (RSVD_0):** Reserved
12:8	0h RW	**Pull Up Override Slew Value (PUOVRSLEWVAL):** Override slew values for pullup and pulldown settings. [4:0] Pull Down Slew Value (nleg) [12:8] Pull Up Slew Value (pleg)
7:5	0h RO	**Reserved (RSVD_1):** Reserved
4:0	0h RW	**Pull Down Override Slew Value (PDOVRSLEWVAL):** Override slew values for pullup and pulldown settings. [4:0] Pull Down Slew Value (nleg) [12:8] Pull Up Slew Value (pleg)

29.1614 RCOMP Value Register DW0 (RCOMP_VALUE_DW0_EMMC_1P8)—Offset 4F0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
24:16	0h RO	**Pull Up Strength Value (STRVALPLEG):** [24:16] Pull Up Strength Value (pleg) [31:25] Reserved
15:9	0h RO	**Reserved (RSVD_1):** Reserved
8:0	0h RO	**Pull Down Strength Value (STRVALNLEG):** [8:0] Pull Down Strength Value (nleg) [15:9] Reserved

29.1615 RCOMP Value Register DW1 (RCOMP_VALUE_DW1_EMMC_1P8)—Offset 4F4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:13	0h RO	**Reserved (RSVD_0):** Reserved
12:8	0h RO	**Pull Up Slew Value (RCOMP_PUSLEWVAL):** Slew values. Valid only after calibration is done. [4:0] Pull Down Slew Value (nleg) [12:8] Pull Up Slew Value (pleg)
7:5	0h RO	**Reserved (RSVD_1):** Reserved
4:0	0h RO	**Pull Down Slew Value (RCOMP_PDSLEWVAL):** Slew values. Valid only after calibration is done. [4:0] Pull Down Slew Value (nleg) [12:8] Pull Up Slew Value (pleg)

29.1616 RCOMP Control Register DW1 (RCOMP_CTL_DW1_EMMC_1P8)—Offset 4F8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 50273h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO	**Reserved (RSVD_0):** Reserved
24	0h RO/V	**RCOMP Calibration Ongoing (RCALON):** 0 = RCOMP FSM is idle 1 = RCOMP calibration is ongoing
23:19	0h RO	**Reserved (RSVD_1):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
18:16	5h RW	**RCOMP Oversampling Count (RCOMP_ROVERSAMPCNT):** 000 Illegal 001 3x DDR sampling 010 5x DDR sampling 011 7x DDR sampling 100 9x DDR sampling 101 11x DDR sampling 110 13x DDR sampling 111 15x DDR sampling
15:13	0h RO	**Reserved (RSVD_2):** Reserved
12:8	2h RW	**RCOMP Aging Activity Factor (RCOMP_RAGEAF):** 00000 -- 0% 00001 -- 5% 00010 -- 10% 00011 -- 15% 00100 -- 20% 00101 -- 25% 00110 -- 30% 00111 -- 35% 01000 -- 40% 01001 -- 45% 01010 -- 50%
7:4	7h RW	**RCOMP Interval Select (RINTSEL):** This field specifies the interval for RCOMP calibration. 0000 - non-periodic mode 0.5ms - 0001 (illegal option) 1ms - 0010 2ms - 0011 3ms - 0100 4ms - 0101 5ms - 0110 10ms - 0111 15ms - 1000 25ms - 1001 50ms - 1010 100ms - 1011 200ms - 1100 500ms - 1101 1000ms - 1110 2000ms - 1111
3	0h RO	**Reserved (RSVD_3):** Reserved
2	0h RW	**halfbit feature enable. (HALFBITEN):** 1: enable halfbit feature 0: disable halfbit feature
1	1h RW	**RCOMP A0 FSM and E0 FSM selection (RCOMP_RFSMSEL):** 1 Choose E0 FSM 0 Choose A0 FSM
0	1h RW	**RCOMP Global Reset (RFSMRST):** RCOMP FSM reset signal, active low.

29.1617 RCOMP Reserved Register (RCOMP_RSVD_EMMC_1P8)—Offset 4FCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** RCOMP reserved register

29.1618 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_176)—Offset 600h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1619 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_176)—Offset 604h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved

Bit Range	Default & Access	Field Name (ID): Description
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1620 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_176)—Offset 608h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1621 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_176)—Offset 60Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1622 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_177)—Offset 610h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)

Bit Range	Default & Access	Field Name (ID): Description
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1623 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_177)—Offset 614h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1624 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_177)—Offset 618h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1625 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_177)—Offset 61Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1626 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_178)—Offset 620h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1627 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_178)—Offset 624h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1628 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_178)—Offset 628h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1629 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_178)—Offset 62Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1630 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_187)—Offset 630h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1631 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_187)—Offset 634h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1632 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_187)—Offset 638h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 100h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	1h RO/V	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1633 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_187)—Offset 63Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1634 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_179)—Offset 640h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1635 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_179)—Offset 644h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1636 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_179)—Offset 648h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1637 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_179)—Offset 64Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1638 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_180)—Offset 650h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1639 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_180)—Offset 654h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1640 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_180)—Offset 658h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1641 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_180)—Offset 65Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1642 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_181)—Offset 660h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1643 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_181)—Offset 664h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1644 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_181)—Offset 668h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1645 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_181)—Offset 66Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1646 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_182)—Offset 670h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1647 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_182)—Offset 674h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1648 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_182)—Offset 678h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1649 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_182)—Offset 67Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1650 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_183)—Offset 680h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1651 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_183)—Offset 684h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1652 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_183)—Offset 688h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1653 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_183)—Offset 68Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1654 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_184)—Offset 690h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1655 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_184)—Offset 694h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1656 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_184)—Offset 698h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1657 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_184)—Offset 69Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1658 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_185)—Offset 6A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1659 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_185)—Offset 6A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1660 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_185)—Offset 6A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1661 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_185)—Offset 6ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1662 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_186)—Offset 6B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1663 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_186)—Offset 6B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1664 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_186)—Offset 6B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1665 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_186)—Offset 6BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1666 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_188)—Offset 6C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1667 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_188)—Offset 6C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1668 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_188)—Offset 6C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1669 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_188)—Offset 6CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1670 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_210)—Offset 6D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000100h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	0h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1671 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_210)—Offset 6D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1672 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_210)—Offset 6D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1673 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_210)—Offset 6DCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1674 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_189)—Offset 6E0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1675 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_189)—Offset 6E4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1676 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_189)—Offset 6E8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1677 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_189)—Offset 6ECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1678 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_190)—Offset 6F0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1679 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_190)—Offset 6F4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1680 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_190)—Offset 6F8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1681 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_190)—Offset 6FCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1682 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_191)—Offset 700h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1683 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_191)—Offset 704h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1684 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_191)—Offset 708h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1685 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_191)—Offset 70Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1686 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_192)—Offset 710h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1687 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_192)—Offset 714h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1688 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_192)—Offset 718h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1689 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_192)—Offset 71Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1690 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_193)—Offset 720h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1691 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_193)—Offset 724h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1692 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_193)—Offset 728h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 800h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	1h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1693 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_193)—Offset 72Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1694 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_194)—Offset 730h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1695 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_194)—Offset 734h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1696 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_194)—Offset 738h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1697 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_194)—Offset 73Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1698 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_195)—Offset 740h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1699 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_195)—Offset 744h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1700 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_195)—Offset 748h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1701 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_195)—Offset 74Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1702 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_196)—Offset 750h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000700h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RW	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1703 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_196)—Offset 754h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	0h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1704 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_196)—Offset 758h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**Reserved (RSVD_0):** Reserved
11	0h RO/V	**Pin-strap value (PINSTRAPVAL):** This bit reflects the pin-strap value.
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1705 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_196)—Offset 75Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1706 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_197)—Offset 760h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000B00h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:12	0h RO	**Reserved (RSVD_1):** Reserved
11:10	2h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	1h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1707 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_197)—Offset 764h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1708 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_197)—Offset 768h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1709 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_197)—Offset 76Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1710 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_198)—Offset 770h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1711 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_198)—Offset 774h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1712 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_198)—Offset 778h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1713 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_198)—Offset 77Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1714 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_199)—Offset 780h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RO	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:11	0h RO	**Reserved (RSVD_1):** Reserved
10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1715 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_199)—Offset 784h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1716 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_199)—Offset 788h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1717 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_199)—Offset 78Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1718 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_200)—Offset 790h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1719 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_200)—Offset 794h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1720 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_200)—Offset 798h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1721 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_200)—Offset 79Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1722 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_201)—Offset 7A0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1723 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_201)—Offset 7A4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1724 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_201)—Offset 7A8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1725 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_201)—Offset 7ACh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1726 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_202)—Offset 7B0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1727 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_202)—Offset 7B4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1728 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_202)—Offset 7B8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1729 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_202)—Offset 7BCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1730 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_203)—Offset 7C0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1731 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_203)—Offset 7C4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1732 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_203)—Offset 7C8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1733 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_203)—Offset 7CCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1734 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_204)—Offset 7D0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1735 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_204)—Offset 7D4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1736 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_204)—Offset 7D8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1737 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_204)—Offset 7DCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1738 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_205)—Offset 7E0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1739 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_205)—Offset 7E4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1740 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_205)—Offset 7E8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1741 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_205)—Offset 7ECh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1742 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_206)—Offset 7F0h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1743 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_206)—Offset 7F4h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1744 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_206)—Offset 7F8h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1745 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_206)—Offset 7FCh

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1746 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_207)—Offset 800h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1747 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_207)—Offset 804h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1748 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_207)—Offset 808h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1749 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_207)—Offset 80Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1750 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_208)—Offset 810h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1751 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_208)—Offset 814h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 3000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	Ch RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1752 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_208)—Offset 818h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1753 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_208)—Offset 81Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

29.1754 Pad Configuration DW0 (PAD_CFG_DW0_GPIO_209)—Offset 820h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 44000500h

Bit Range	Default & Access	Field Name (ID): Description
31:30	1h RW	**Pad Reset Config (PADRSTCFG):** This register controls which reset is used to reset the specified Pad Register fields. It is the SoC responsibilities to decide the number of configurable reset required with an example below: 00 = Powergood (i.e. sticky reset) 01 = Deep GPIO Reset (i.e. host deep reset: host_deep_rst_b) 10 = GPIO Reset (i.e. host reset: host_rst_b) 11 = Reserved; tied to inactive (i.e. '1' for active low reset) For DSW well pads, this register default to 00b. For Primary well GPIO pads, this register default to 01b Note: The hardware must ensure that when this field is programmed, the muxed reset signal does not glitch to cause momentarily reset assertion.
29	0h RW	**RX Pad State Select (RXPADSTSEL):** Determine from which node the RX pad state for native function should be taken from. This field only affects the pad state value being fanned out to native function(s) and is not meaningful if the pad is in GPIO mode (i.e. Pad Mode = 0). 0 = Raw RX pad state directly from CFIO RX buffer 1 = Internal RX pad state (subject to RXINV , hardware debouncer (if any) and PreGfRXSel settings)
28	0h RW	**RX Raw Override to '1' (RXRAW1):** This bit determines if the selected pad state is being overridden to '1'. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The override takes place at the internal pad state directly from buffer and before the RXINV. 0 = No Override 1 = RX drive 1 internally
27	0h RO	**Reserved (RSVD_0):** Reserved
26:25	2h RW	**RX Level/Edge Configuration (RXEVCFG):** Determine if the internal RX pad state (synchronized, filtered vs non-filtered version as determined by PreGfRXSel, and is further subject to RXInv) should be passed on to the next logic stage as is, as a pulse, or level signal. This field does not affect the received pad state (to GPIORXState or native functions) but how the interrupt or wake triggering events should be delivered to the GPIO Community Controller . 0h = Level 1h = Edge 2h = Drive '0' 3h = Either rising edge or failing edge
24	0h RO	**Pre Glitch Filter Stage RX Pad State Select (PREGFRXSEL):** Determine if the synchronized version of the raw RX pad state should be subjected to glitch filter or not. This field only makes sense when the RX buffer is configured as an input and is not disabled. 0 = Select synchronized, non filtered RX pad state 1 = Select synchronized, filtered RX pad state The selected RX pad state can be further subjected to polarity inversion through RXINV
23	0h RW	**RX Invert (RXINV):** This bit determines if the selected pad state should go through the polarity inversion stage. This field only makes sense when the RX buffer is configured as an input in either GPIO Mode or native function mode. The polarity inversion takes place at the mux node of raw vs filtered or non-filtered RX pad state, as determined by PreGfRXsel and RXPadStSel This bit does not affect GPIORXState. During host ownership GPIO Mode, when this bit is set to '1', then the RX pad state is inverted as it is sent to the GPIO-to-IOxAPIC, GPE/SCI, SMI, NMI logic or GPI_IS[n] that is using it. This is used to allow active-low and active-high inputs to cause IRQ, SMI#, SCI or NMI. 0 = No inversion 1 = Inversion
22:21	0h RO	**RX/TX Enable Config (RXTXENCFG):** This controls the RX and TX buffer enables for the function selected by Pad Mode, but is not applicable when Pad Mode is 0 (i.e. GPIO mode). Hardware shall ensure GPIOTxDis and GPIORxDis are controlling the RX and TX buffers when this field is 0 and Pad Mode is 0. 0 = Function defined in Pad Mode controls TX and RX Enables 1 = Function controls TX Enable and RX Disabled with RX drive 0 internally 2 = Function controls TX Enable and RX Disabled with RX drive 1 internally 3 = Function controls TX Enabled and RX is always enabled

Bit Range	Default & Access	Field Name (ID): Description
20	0h RO	**GPIO Input Route IOxAPIC (GPIROUTIOXAPIC):** Determine if the pad can be routed to cause peripheral IRQ when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause peripheral IRQ 1 = Routing can cause peripheral IRQ Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the peripheral IRQ indication to the intended recipient(s).
19	0h RW	**GPIO Input Route SCI (GPIROUTSCI):** Determine if the pad can be routed to cause SCI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SCI 1 = Routing can cause SCI Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the GPE indication to the intended recipient(s), e.g. PMU.
18	0h RO	**GPIO Input Route SMI (GPIROUTSMI):** Determine if the pad can be routed to cause SMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause SMI. 1 = Routing can cause SMI. Note: This bit does not affect any interrupt status bit within GPIO, but is used as the last qualifier for the SMI indication to the intended recipient(s).
17	0h RO	**GPIO Input Route NMI (GPIROUTNMI):** Determine if the pad can be routed to cause NMI when configured in GPIO input mode. If the pad is not configured in GPIO input mode, this field has no effect. 0 = Routing does not cause NMI. 1 = Routing can cause NMI. Note: This bit also affects GPI_NMI_STS. If '0', GPI_NMI_STS is always clear. If '1', GPI_NMI_STS could be set (depending on GPIOOwn setting) when there is an event. Whether a NMI indication is generated and sent to the intended recipient(s) is also depending on the corresponding GPI_NMI_EN bit.
16:14	0h RO	**Reserved (RSVD_1):** Reserved
13:10	1h RW	**Pad Mode (PMODE):** This field determines whether the Pad is controlled by GPIO controller logic or one of the native functions muxed onto the Pad. 0h = GPIO control the Pad 1h = Function 1, if applicable, control the Pad ... 15 = Function 15 control the Pad
9	0h RW	**GPIO RX Disable (GPIORXDIS):** 0 = Enable the input buffer (active low enable) of the pad 1 = Disable the input buffer of the pad. Notes: When the input buffer is disabled, the internal pad state is always driven to '0' by CFIO HIP.
8	1h RW	**GPIO TX Disable (GPIOTXDIS):** 0 = Enable the output buffer (active low enable) of the pad 1 = Disable the output buffer of the pad; i.e. Hi-Z
7:2	0h RO	**Reserved (RSVD_2):** Reserved
1	0h RO/V	**GPIO RX State (GPIORXSTATE):** This is the current internal RX pad state after Glitch Filter logic stage and hardware debouncer (if any) and is not affected by PMode and RXINV settings.
0	0h RW	**GPIO TX State (GPIOTXSTATE):** 0 = Drive a level '0' to the TX output pad 1 = Drive a level '1' to the TX output pad

29.1755 Pad Configuration DW1 (PAD_CFG_DW1_GPIO_209)—Offset 824h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 1000h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RO	**CFIO Pad Configuration I2CEN (CFIOPADCFG_I2CEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
29	0h RO	**CFIO Pad Configuration CSEN (CFIOPADCFG_CSEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
28	0h RO	**CFIO Pad Configuration PARKMODEEN_B (CFIOPADCFG_PARKMODEEN_B):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
27:26	0h RO	**CFIO Pad Configuration HYSCTL (CFIOPADCFG_HYSCTL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
25	0h RO	**CFIO Pad Configuration PADTOL (CFIOPADCFG_PADTOL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
24:22	0h RO	**CFIO Pad Configuration STRSEL (CFIOPADCFG_STRSEL):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
21	0h RO	**CFIO Pad Configuration HSMODE (CFIOPADCFG_HSMODE):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
20	0h RO	**CFIO Pad Configuration ODTUPDN (CFIOPADCFG_ODTUPDN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
19	0h RO	**CFIO Pad Configuration ODTEN (CFIOPADCFG_ODTEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.

Bit Range	Default & Access	Field Name (ID): Description
18	0h RO	**CFIO Pad Configuration ANALOGMUXEN (CFIOPADCFG_ANALOGMUXEN):** See CFIO HIP Interface for details on these control bit descriptions. These may be duplicates in the family settings register since these can be either Family or Pad signals and this is SoC specific. The superset is described here and register bits are implemented only for those configurations that are pad specific. Bits that aren't implemented or output of the CFIO are read only.
17:14	0h RW	**IO Standby State (IOSSTATE):** The I/O Standby State defines which state the pad should be parked in when the I/O is in a standby state. 0 = Latch last value driven on TX, TX Enable and RX Enable 1 = Drive 0 with RX disabled and RX drive 0 internally 2 = Drive 0 with RX disabled and RX drive 1 internally 3 = Drive 1 with RX disabled and RX drive 0 internally 4 = Drive 1 with RX disabled and RX drive 1 internally 5 = Drive 0 with RX enabled 6 = Drive 1 with RX enabled 7 = Hi-Z with RX drive 0 internally 8 = Hi-Z with RX drive 1 internally 9 = TX Disabled and RX Enabled (i.e. wake or interrupt) 15 = IO Standby signal is masked for this pad. In this mode, a pad operates as if IOStandby has not been asserted. Others = Reserved
13:10	4h RW	**Termination (TERM):** The Pad Termination state defines the different weak pull-up and pull-down settings that are supported by the buffer. The settings for [13:10] correspond to: 0 000: none 0 010: 5k wpd 0 100: 20k wpd 1 000: none 1 001: 1k wpu 1 011: 2k wpu 1 010: 5k wpu 1 100: 20k wpu 1 101: 1k & 2k wpu 1 111: (optional) Native controller selected by Pad Mode controls the Termination All others reserved. If a reserved value is programmed, pad may malfunction. The setting of this field is applicable in all Pad Mode including GPIO. As each Pad Mode may require different termination and isolation, care must be taken in sw/fw in the transition with appropriate register programming. The actual transition sequence requirement may vary on case by case basis depending on the native functions involved. For example, before changing the pad from output to input direction, wpu/wpd settings should be programmed first to ensure the input does not float momentarily.
9:8	0h RW	**IO Standby Termination (IOSTERM):** IOSTerm defines the behaviour of the termination logic when the IO Standby state has been triggered. The value of the termination is determined in Term[2:0]. 0 = Same as state specified in Term. 1 = Disable Pullup and Pulldown 2 = Enable Pulldown 3 = Enable Pullup
7:0	0h RO	**Interrupt Select (INTSEL):** The Interrupt Select defines which interrupt line driven from the GPIO Controller toggles when an interrupt is detected on this pad. 0 = Interrupt Line 0 1 = Interrupt Line 1 Up to the max IOxAPIC IRQ supported

29.1756 Pad Configuration DW2 (PAD_CFG_DW2_GPIO_209)—Offset 828h

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RO	**Reserved (RSVD_0):** Reserved
10	0h RO	**Bus Hold Enable (BUSHLDEN):** 1 = enable bus hold 0 = disable bus hold
9	0h RO	**Reserved (RSVD_1):** Reserved
8	0h RO	**VCCIO selection (VCCIOSEL):** This bits is used to select VCCIO If the pad supports programmable VCCIO. The definition of high and low are implementation specific. 0: low voltage 1: high voltage
7:5	0h RO	**Reserved (RSVD_2):** Reserved
4:1	0h RW	**Debounce duration (DEBOUNCE):** The debounce duration of value should be not less than 3. In other words, the value should not be any of 0, 1 and 2. The debounce time should be calculated as (2^debounce) * (glitch filter clock period) For example, when RTC clock (32 KHz) is used for glitch filter and debounce duration is set to 4b1011, the debounce time is (2 ^ 11) * (31.25 us) = 64ms
0	0h RW	**Debounce Enable (DEBEN):** This bit enables or disables the debouncer with the pad. 1 = Enable the debouncer of the pad 0 = Disable the debouncer of the pad

29.1757 Pad Configuration DW3 (PAD_CFG_DW3_GPIO_209)—Offset 82Ch

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**rsvd (RSVD):** PAD CFG reserved register DW3

§ §

30 MSR Registers

30.1 IA32_MC5_CTL (P_CR_MC_CTL)—Offset 4620h

PUNIT_MSR: PCU Machine Check Architecture Control MSR.

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 7h

Bit Range	Default & Access	Field Name (ID): Description
63:3	0h RO/V	**RESERVED_0 (RESERVED_0):** Reserved
2	1h RW	**EN_PCU_FW_ERR (EN_PCU_FW_ERR):** Enable Error signaling of PCU Pcode errors. When bit is set PCU HW errors will result in assertion of CATERR. Otherwise CATERR is not asserted. In any case the error is logged in MC4_STATUS MSR.
1	1h RW	**EN_PCU_UC_ERR (EN_PCU_UC_ERR):** Enable Error signaling of PCU uC errors. When bit is set PCU HW errors will result in assertion of CATERR. Otherwise CATERR is not asserted. In any case the error is logged in MC4_STATUS MSR.
0	1h RW	**EN_PCU_HW_ERR (EN_PCU_HW_ERR):** Enable Error signaling of PCU HW errors. When bit is set PCU HW errors will result in assertion of CATERR. Otherwise CATERR is not asserted. In any case the error is logged in MC4_STATUS MSR.

30.2 IA32_MC5_STATUS (P_CR_MC_STATUS)—Offset 4628h

PUNIT_MSR: PCU Machine Check Architecture Status MSR.

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW/V	**VALID (VALID):** Indicates that this register contains a valid MCA error that was detected.
62	0h RW/V	**OVR (OVR):** Another MCA error was detected when VALID was already set to 1b. In other words a second MCA error was detected while an earlier error was already logged.
61	0h RW/V	**UC (UC):** Indicates when set that the processor did not correct the error condition. 0b Correctable Error 1b Uncorrectable Error
60	0h RW/V	**EN (EN):** Enable bit Whether the logged error was enabled or not. This bit is equal to the appropriate enable bit in IA32_MC4_CTL: For HW errors the value of IA32_MC4_CTL[EN_PCU_HW_ERROR] is writen by HW to this bit. For Microcontroller errors the value of IA32_MC4_CTL[EN_PCU_UC_ERROR] is writen by HW to this bit. For FW errors the value of IA32_MC4_CTL[EN_PCU_FW_ERROR] is writen by HW to this bit.

Bit Range	Default & Access	Field Name (ID): Description
59	0h RW/V	**MISCV (MISCV):** The value held in the MC4_MISC_MSR is valid
58	0h RW/V	**ADDRV (ADDRV):** The address held in the MC4_ADDR_MSR is valid
57	0h RW/V	**PCC (PCC):** Processor Context Corrupt
56:55	0h RW/V	**ENH_MCA_AVAIL1 (ENH_MCA_AVAIL1):** Available when Enhanced MCA is in use
54:53	0h RW/V	**CORRERRORSTATUSIND (CORRERRORSTATUSIND):** These bits are used to indicate when the number of corrected errors has exceeded the safe threshold to the point where an uncorrected error has become more likely to happen. 00 no Tracking No Hardware status tracking is provided for the structure reporting this event. Can I assume that status is 00 when the SBF is disabled 01 Green Status Tracking is provided for the structure reporting this event Current status is Green indicating that the structure is healthy. This is the status when the SBF is not disabled and not full 10 Yellow Status Tracking is provided for the structure reporting this event Current status is Yellow indicating that the structure is operating correctly but with degraded status. This is the status when the SBF is not disabled but the number of HARD errors is above certain threshold defined in CBO_MCA_CONFIG CR 11 Reserved
52:38	0h RW/V	**CORR_ERR_COUNT (CORR_ERR_COUNT):** Correctable error count. Not exposed for PCU
37:32	0h RW/V	**ENH_MCA_AVAIL0 (ENH_MCA_AVAIL0):** Available when Enhanced MCA is in use
31:24	0h RW/V	**MSEC_FW (MSEC_FW):** Model specific error code. This value is written by pcode or vcode from IO_Firmware_MCA_Command. It contains the pcode/vcode specific failure that was logged. Exact encodings found in: machine_check_defs.finc
23:20	0h RW/V	**MSEC_HW (MSEC_HW):** Model specific error code. This logs the type of HW FSM error that has occured. There are 3 errors defined. 1. C7 Save/Restore FSM ECC Error on FSM A 2. C7 Save/Restore FSM ECC Error on FSM B 3. BCLK Mode FSM Failure
19:16	0h RW/V	**MSEC_UC (MSEC_UC):** Model specific error code. This logs the type of HW uC PCU error that has occured. There are 7 errors defined. 1. Instruction address out of valid space 2. Double bit RAM error on Instruction Fetch 3. Invalid OpCode seen Any colored line in the excel sheet 4. Stack Underflow 5. Stack Overflow 6. Data address out of valid space 7. Double bit RAM error on Data Fetch
15:0	0h RW/V	**MCCOD (MCCOD):** Machine Check Error Code. All PCU errors are classified as Internal Unclassified Errors. The value of this field will be 0402h for the PCU . This applies for any logged error

30.3 IA32_MC5_CTL2 (P_CR_MC_CTL2)—Offset 4630h

PUNIT_MSR: PCU Machine Check Architecture Control MSR.

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:35	0h RO	**RESERVED_3 (RESERVED_3):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
34	0h RW	**MCE_CTL (MCE_CTL):** Controls MCE Remapping 0 MCE 1 MSMI
33	0h RO	**RESERVED_2 (RESERVED_2):** Reserved
32	0h RW	**CMCI_CTL (CMCI_CTL):** Controls CMCI Remapping 0 CMCI 1 CSMI
31	0h RO	**RESERVED_1 (RESERVED_1):** Reserved
30	0h RO	**ENABLE (ENABLE)**
29:15	0h RO	**RESERVED_0 (RESERVED_0):** Reserved
14:0	0h RO	**THRESHOLD (THRESHOLD)**

30.4 VR_MISC_CONFIG (P_CR_VR_MISC_CONFIG)—Offset 4670h

PUNIT_MSR: Input voltage regulator configuration parameters. PCODE will read this register during Reset Phase 4. **MSR_Name**: VR_MISC_CONFIG **MSR_Addr**: 0x603

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 8FF0Bh

Bit Range	Default & Access	Field Name (ID): Description
63:32	0h RO	Reserved.
31:20	0h RO	**RSVD3 (RSVD3):** Reserved Bits
19:18	2h RW	**SLOW_SLEW_RATE_CONFIG_rail1 (SLOW_SLEW_RATE_CONFIG_RAIL1):** Defines the desired ratio between the VR fast slope and slow slope for Vnn.
17:16	0h RO	**RSVD2 (RSVD2):** Reserved Bits
15:8	FFh RW	**FAST_RAMP_VOLTAGE_RAIL0 (FAST_RAMP_VOLTAGE_RAIL0):** The maximum voltage that the CPU can ramp at the fast rate to
7:4	0h RO	**RSVD1 (RSVD1):** Reserved Bits
3:2	2h RW	**SLOW_SLEW_RATE_CONFIG_rail0 (SLOW_SLEW_RATE_CONFIG_RAIL0):** Defines the desired ratio between the VR fast slope and slow slope for Vcc.
1	1h RW	**IDLE_ENTRY_RAMP_RATE_RAIL0 (IDLE_ENTRY_RAMP_RATE_RAIL0):** This bit controls the VR voltage slew rate on package Cstate entry events. • 1 = slow • 0 = decay

Bit Range	Default & Access	Field Name (ID): Description
0	1h RW	**IDLE_EXIT_RAMP_RATE_RAIL0 (IDLE_EXIT_RAMP_RATE_RAIL0):** This bit controls the VR voltage slew rate on package Cstate wake events. • 1 = fast • 0 = slow

30.5 Thread T-state Request (P_CR_THREAD_T_REQ)—Offset 46B8h

PUNIT_MSR: Interface for SW controlled clock modulation. Mechanism is implemented in GLM using Ehalt. **MSR_Name**: IA32_CLOCK_MODULATION **MSR_Addr**: 0x19A THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RO	Reserved.
4	0h RW	**THROTTLE_ACTIVE (THROTTLE_ACTIVE):** Enables on-demand software controlled clock modulation.
3:0	0h RW	**T_STATE_REQ (T_STATE_REQ):** Selects the clock modulation duty cycle (in 6.45% increments). This field is only active when the clock modulation enable flag is set (THROTTLE_ACTIVE).

30.6 IA32_MC5_ADDR (P_CR_MC_ADDR)—Offset 46E0h

PUNIT_MSR: PCU Machine Check Architecture Address MSR.

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:0	0h RW/V	**ENH_MCA_AVAIL (ENH_MCA_AVAIL):** Available when enhanced MCA is in use

30.7 IA32_MC5_MISC (P_CR_MC_MISC)—Offset 46E8h

PUNIT_MSR: PCU Machine Check Architecture Misc MSR.

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:32	0h RW/V	**ENH_MCA_AVAL (ENH_MCA_AVAL):** Available when enhanced MCA is in use
31:24	0h RW/V	**CORRECTED_ERR_CNT (CORRECTED_ERR_CNT):** This field contains the count of correctable errors seen in the PCU. It is updated on a correctable error.
23:17	0h RW/V	**ENH_MCA_AVAL1 (ENH_MCA_AVAL1):** Available when enhanced MCA is in use
16:0	0h RW/V	**ERROR_ADDRESS (ERROR_ADDRESS):** This field contains the address of uncorrected error seen in the PCU. It is updated on an uncorrectable error.

30.8 PLATFORM_ID (P_CR_PLATFORM_ID)—Offset 4710h

Indicates the platform the porcessor is intended for and is also used for selecting the correct patch to apply from among those available in BIOS. Accessible by BIOS as an MSR.

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:53	0h RSV	**RESERVED_1 (RESERVED_1):** Reserved
52:50	0h RW/V	**PLATFORM_ID (PLATFORM_ID):** This field is loaded from fuses. The field gives information concerning the intended platform for the processor.
49:0	0h RSV	**RESERVED_0 (RESERVED_0):** Reserved

30.9 PLATFORM_INFO (P_CR_PLATFORM_INFO)—Offset 4718h

PUNIT_MSR: This contains information about platforms frequency capabilities that the CPU reports. **MSR_Name**: PLATFORM_INFO **MSR_Addr**: 0x0CE

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 10000F0010000h

Bit Range	Default & Access	Field Name (ID): Description
63:56	0h RO	**RESERVED_5 (RESERVED_5):** reserved
55:48	1h RW	**Minimum Supported Clock Multiplier (MIN_OPERATING_RATIO):** Mimumum supported clock multiplier for IA cores, described in units of 100MHz. This is the lower bound on what software may program into its P-state request for IA cores
47:40	0h RW	**Maximum Efficiency Clock Multiplier (MAX_EFFICIENCY_RATIO):** Maximum Efficienty Ratio. This is given in units of 100 MHz.
39:38	0h RO	**RESERVED_4 (RESERVED_4):** reserved
37	0h RW	**Timed MWAIT Enable (TIMED_MWAIT_ENABLE):** If set, this field indicates that the core Timed WAIT feature is supported.
36	0h RO	**RESERVED_3 (RESERVED_3):** reserved
35	0h RW	**PFAT_ENABLE (PFAT_ENABLE):** If set, this field indicates that PFAT is enabled
34:33	0h RW	**CONFIG_TDP_LEVELS (CONFIG_TDP_LEVELS):** Indicates the number of configurable TDP levels are supported. If set to zero, configurable TDP is not supported as a feature
32	0h RW	**Low Power Mode Support (LPM_SUPPORT):** Indicates whether or not P-states below the maximally efficient P-state are allowed. All products support LPM.
31	1h RW	**CPUID Faulting Enable (CPUID_FAULTING_EN):** CPUID Faulting
30	1h RW	**Programmable Tj Offset Enable (PRG_TJ_OFFSET_EN):** If set, enables programmable Tj offset in the TEMPERATURE_TARGET.TJ_MAX_TCC_OFFSET field.
29	1h RW	**Programmable TDP Limit Enable (PRG_TDP_LIMIT_EN):** If set, PL1 power limits are programmable. If clear, the PL1 power limit may not be programmed by software
28	1h RW	**Programmable Turbo Ratio Limits Enable (PRG_TURBO_RATIO_EN):** If set, the FREQ_LIMIT_RATIOS register is programmable by software. If clear, this register is not writable
27	0h RW	**Sample Part (SAMPLE_PART):** Indicates that this part is a non-production / sample part
26	0h RW	**DCU_16K_MODE_AVAIL (DCU_16K_MODE_AVAIL):** If set, indicates the part supports 16K DCU mode
25	0h RO	**RESERVED_2 (RESERVED_2):** Reserved
24	0h RW	**Overclocking Voltage Override Enable (OCVOLT_OVRD_EN):** If set, indicates the part supports voltage overrides as part of general overclocking and voltage control features
23	0h RW	**PPIN_CAP (PPIN_CAP):** When set to 1, indicates that Protected Processor Inventory Number (PPIN) capability can be enabled for privileged system inventory agent to read PPIN from MSR_PPIN. When set to 0, PPIN capability is not supported. An attempt to access MSR_PPIN_CTL or MSR_PPIN will cause #GP.
22:17	0h RO	**RESERVED_1 (RESERVED_1):** Reserved
16	1h RO	**SMM_SAVE_CAP (SMM_SAVE_CAP):** Save/Restore x87 state in SMM Reserved
15:8	0h RW	**Maximum Non-Turbo Clock Multiplier (MAX_NON_TURBO_LIM_RATIO):** The Maximum Non-Turbo clock ratio multiplier describes the highest clock frequency in 100MHz units that IA cores are manufactured to run at without hitting thermal or power constraints. This clock frequency also defines the level that the core timestamp counter runs at

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RO	**RESERVED_0 (RESERVED_0):** Reserved

30.10 DDR_RAPL_LIMIT (P_CR_DDR_RAPL_LIMIT)—Offset 4760h

RAPL power limit for DDR domain. This register is written by platform software and read by pcode once per 1ms. There are actually two instances of this register: one with MSR access and one with MMIO access. There is a separate PECI/PCS command which is also analogous.

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW/L	**LOCKED (LOCKED):** When set this entire register becomes readonly. This bit will typically be set by BIOS during boot.
62:32	0h RO	Reserved.
31:24	0h RSV	**RESERVED_1 (RESERVED_1):** Reserved
23:22	0h RW/L	**LIMIT1_TIME_WINDOW_X (LIMIT1_TIME_WINDOW_X):** The time window for RAPL is expressed in a format called TAU. TAU is specified in a mantissa+exponent format. This format is common across all RAPL interfaces for TAU. The format is specified with a 7 bit number that looks like: XXYYYYY (2 bits of mantissa follwed by 5 exponent) The time window corresponding to this config is calculated as: (1+ XXb / 4) * (2^YYYYYb) The value that results from this math species the time window in units that are defined in PACKAGE_POWER_SKU_UNIT.TIME_UNIT MSR. On every server product we have built, this value has been 976us, which we typically treat as 1ms.
21:17	0h RW/L	**LIMIT1_TIME_WINDOW_Y (LIMIT1_TIME_WINDOW_Y):** The Tau Exponent. See TIME_WINDOW_X for more information.
16	0h RSV	**RESERVED_0 (RESERVED_0):** Reserved
15	0h RW/L	**LIMIT1_ENABLE (LIMIT1_ENABLE):** Power Limit[0] enable bit for DDR domain.
14:0	0h RW/L	**LIMIT1_POWER (LIMIT1_POWER):** Power Limit[0] for DDR domain. UnitsWatts Format7.8 Resolution0.125W Range02047.875W.

30.11 FREQ_LIMIT_RATIOS (P_CR_FREQ_LIMIT_RATIOS)—Offset 4768h

PUNIT_MSR: In conjunction with FREQ_LIMIT_CORES, this MSR allows an end-user to limit the number of turbo bins for different numbers of active cores (cores not in C6). Each field in FREQ_LIMIT_RATIOS specifies a frequency limit for the specified number of active cores in the corresponding field in FREQ_LIMIT_CORES.
Example: RATIO_LIMIT_0 is 0x20, CORE_COUNT_0 is 0x3; RATIO_LIMIT_1 is 0x16 and

CORE_COUNT_1 is 0x5; RATIO_LIMIT_3 is 0x10, CORE_COUNT_3 is 0xFF; The frequency limit of <=3 active cores is 32, the limit for 4-5 active cores is 22, and the limit for 6 or more active cores is 16
Ratio limits must monotonically decrease in ascending bin order, core count must monotonically increase in ascending bin order, and the last valid FREQ_LIMIT_CORE field must be >= number of potentially active cores (just set to 0xFF)
Values of '0' are ignored. **MSR_Name**: FREQ_LIMIT_RATIOS **MSR_Addr**: 0x1AD

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:56	0h RW	**RATIO_LIMIT_7 (RATIO_LIMIT_7):** The frequency limit corresponding to the maximum number of active cores specified in CORE_COUNT_7 in FREQ_LIMIT_CORES.
55:48	0h RW	**RATIO_LIMIT_6 (RATIO_LIMIT_6):** The frequency limit corresponding to the maximum number of active cores specified in CORE_COUNT_6 in FREQ_LIMIT_CORES.
47:40	0h RW	**RATIO_LIMIT_5 (RATIO_LIMIT_5):** The frequency limit corresponding to the maximum number of active cores specified in CORE_COUNT_5 in FREQ_LIMIT_CORES.
39:32	0h RW	**RATIO_LIMIT_4 (RATIO_LIMIT_4):** The frequency limit corresponding to the maximum number of active cores specified in CORE_COUNT_4 in FREQ_LIMIT_CORES.
31:24	0h RW	**RATIO_LIMIT_3 (RATIO_LIMIT_3):** The frequency limit corresponding to the maximum number of active cores specified in CORE_COUNT_3 in FREQ_LIMIT_CORES.
23:16	0h RW	**RATIO_LIMIT_2 (RATIO_LIMIT_2):** The frequency limit corresponding to the maximum number of active cores specified in CORE_COUNT_2 in FREQ_LIMIT_CORES.
15:8	0h RW	**RATIO_LIMIT_1 (RATIO_LIMIT_1):** The frequency limit corresponding to the maximum number of active cores specified in CORE_COUNT_1 in FREQ_LIMIT_CORES.
7:0	0h RW	**RATIO_LIMIT_0 (RATIO_LIMIT_0):** The frequency limit corresponding to the maximum number of active cores specified in CORE_COUNT_0 in FREQ_LIMIT_CORES.

30.12 FREQ_LIMIT_CORES (P_CR_FREQ_LIMIT_CORES)—Offset 4770h

PUNIT_MSR: In conjunction with FREQ_LIMIT_RATIOS, this MSR allows an end-user to limit the number of turbo bins for different numbers of active cores (cores not in C6). Each field in FREQ_LIMIT_RATIOS specifies a frequency limit for the specified number of active cores in the corresponding field in FREQ_LIMIT_CORES.
Example: RATIO_LIMIT_0 is 0x20, CORE_COUNT_0 is 0x3; RATIO_LIMIT_1 is 0x16 and CORE_COUNT_1 is 0x5; RATIO_LIMIT_3 is 0x10, CORE_COUNT_3 is 0xFF; The frequency limit of <=3 active cores is 32, the limit for 4-5 active cores is 22, and the limit for 6 or more active cores is 16
Ratio limits must monotonically decrease in ascending bin order, core count must monotonically increase in ascending bin order, and the last valid FREQ_LIMIT_CORE field must be >= number of potentially active cores (just set to 0xFF)
Values of '0' are ignored. **MSR_Name**: FREQ_LIMIT_CORES **MSR_Addr**: 0x1AE

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:56	0h RW	**CORE_COUNT_7 (CORE_COUNT_7):** The number of maximum active cores corresponding to the frequency limit specified in RATIO_LIMIT_7 in FREQ_LIMIT_RATIOS
55:48	0h RW	**CORE_COUNT_6 (CORE_COUNT_6):** The number of maximum active cores corresponding to the frequency limit specified in RATIO_LIMIT_6 in FREQ_LIMIT_RATIOS
47:40	0h RW	**CORE_COUNT_5 (CORE_COUNT_5):** The number of maximum active cores corresponding to the frequency limit specified in RATIO_LIMIT_5 in FREQ_LIMIT_RATIOS
39:32	0h RW	**CORE_COUNT_4 (CORE_COUNT_4):** The number of maximum active cores corresponding to the frequency limit specified in RATIO_LIMIT_4 in FREQ_LIMIT_RATIOS
31:24	0h RW	**CORE_COUNT_3 (CORE_COUNT_3):** The number of maximum active cores corresponding to the frequency limit specified in RATIO_LIMIT_3 in FREQ_LIMIT_RATIOS
23:16	0h RW	**CORE_COUNT_2 (CORE_COUNT_2):** The number of maximum active cores corresponding to the frequency limit specified in RATIO_LIMIT_2 in FREQ_LIMIT_RATIOS
15:8	0h RW	**CORE_COUNT_1 (CORE_COUNT_1):** The number of maximum active cores corresponding to the frequency limit specified in RATIO_LIMIT_1 in FREQ_LIMIT_RATIOS
7:0	0h RW	**CORE_COUNT_0 (CORE_COUNT_0):** The number of maximum active cores corresponding to the frequency limit specified in RATIO_LIMIT_0 in FREQ_LIMIT_RATIOS

30.13 Package RAPL Power Limit (P_CR_PACKAGE_RAPL_LIMIT)—Offset 4780h

Package RAPL Power Limit allows a software agent to define power limitation for the package domain. Power limitation is defined in terms of average power usage (Watts) over a time window specified. Two power limits and associated time windows can be specified. These power limits are commonly referred to as PL1 (long time window) and PL2 (short time window). Each power limit provides independent clamping control that would permit the processor cores to go below OS-requested state to meet the power limits. A lock mechanism allow the software agent to enforce power limit settings. Once the lock bit is set, the power limit settings are static and un-modifiable until next RESET.

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW/L	**Package RAPL Lock (PKG_PWR_LIM_LOCK):** When set all settings in this register are locked and are treated as Read Only. This lock control is persistent until the next reset. This bit will typically set by BIOS during boot time or resume from Sx.
62:56	0h RSV	**RESERVED_1 (RESERVED_1):** Reserved
55:49	0h RW/L	**Power Limit 2 (PL2) Time Window (PKG_PWR_LIM_2_TIME):** Time window for Power Limit 1 (PL2). This describes the control window of the power limit. This time window is described in an RC time constant format, which means that if 1s is programmed, the power limit constraint really applies at more like 5s. The maximal time window is bounded by PACKAGE_POWER_SKU_MSR.PKG_MAX_WIN. Thre is no constraint on the minimum programmable time window, however at very short time windows the control algorithms may not be effective. The bits of this field describe parameters for a mathematical equation for time window configuration. This field is split into two sub-fields: • x = bits[6:5] • y = bits[4:0] Time window equation: **time_window = PACKAGE_POWER_SKU_UNIT.TIME_UNIT * ((1+x/4)^y)**
48	0h RW/L	**Power Limit 2 (PL2) Clamp (PKG_CLMP_LIM_2):** Clamp mode control for PL2. • 0 = PL2 power control is prevented from forcing P-states below the base frequency / P1 for any domain in the SOC. • 1 = PL2 power control will take all actions necessary to meet the power target, even if that involves running at clock frequencies below the base frequency / P1 level. In order to ensure proper SOC cooling, it is generally recommended that the clamp mode is always enabled.
47	0h RW/L	**Power Limit 2 (PL2) Enable (PKG_PWR_LIM_2_EN):** Enable for Power Limit 2 (PL2). Setting this bit activates the power limit and time window defined for PL2.
46:32	0h RW/L	**Power Limit 2 (PL2) (PKG_PWR_LIM_2):** Sets the average power usage limit of the package domain corresponding to the PL2 time window. The power units of this field are specified by the PACKAGE_POWER_SKU_UNIT_MSR.PWR_UNIT. This power limit must be configured by software before it will engage. The PL2 limit is most commonly associated with long time windows (1s and longer), although there are no explicit constraints on what software configures.
31:24	0h RSV	**RESERVED_0 (RESERVED_0):** Reserved
23:17	0h RW/L	**Power Limit 1 (PL1) Time Window (PKG_PWR_LIM_1_TIME):** Time window for Power Limit 1 (PL1). This describes the control window of the power limit. This time window is described in an RC time constant format, which means that if 1s is programmed, the power limit constraint really applies at more like 5s. The maximal time window is bounded by PACKAGE_POWER_SKU_MSR.PKG_MAX_WIN. Thre is no constraint on the minimum programmable time window, however at very short time windows the control algorithms may not be effective. The bits of this field describe parameters for a mathematical equation for time window configuration. This field is split into two sub-fields: • x = bits[6:5] • y = bits[4:0] Time window equation: **time_window = PACKAGE_POWER_SKU_UNIT.TIME_UNIT * ((1+x/4)^y)**
16	0h RW/L	**Power Limit 1 (PL1) Clamp (PKG_CLMP_LIM_1):** Clamp mode control for PL1. • 0 = PL1 power control is prevented from forcing P-states below the base frequency / P1 for any domain in the SOC. • 1 = PL1 power control will take all actions necessary to meet the power target, even if that involves running at clock frequencies below the base frequency / P1 level. In order to ensure proper SOC cooling, it is generally recommended that the clamp mode is always enabled.
15	0h RW/L	**Power Limit 1 (PL1) Enable (PKG_PWR_LIM_1_EN):** Enable for Power Limit 1 (PL1). Setting this bit activates the power limit and time window defined for PL1.

Bit Range	Default & Access	Field Name (ID): Description
14:0	0h RW/L	**Power Limit 1 (PL1) (PKG_PWR_LIM_1):** Sets the average power usage limit of the package domain corresponding to the PL1 time window. The power units of this field are specified by the PACKAGE_POWER_SKU_UNIT_MSR.PWR_UNIT. This power limit must be configured by software before it will engage. The PL1 limit is most commonly associated with long time windows (1s and longer), although there are no explicit constraints on what software configures.

30.14 PL3_CONTROL_MSR (P_CR_PL3_CONTROL)—Offset 4790h

PUNIT_MSR: PL3 control MSR

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW/L	**LOCK (LOCK):** Lock this MSR until next reset: 0 - unlocked, 1 - locked
62:48	0h RO	Reserved.
47	0h RW/L	**PL4_ENABLE (PL4_ENABLE):** 0 => Algorithm disabled, 1 => Algorithm enabled
46:32	0h RW/L	**PMAX (PMAX):** Power Limit 'PL4' or Pmax power limit in the units as described PACAKGE_POWER_SKU_UNIT MSR
31	0h RO	Reserved.
30:24	0h RW/L	**DUTY_CYCLE (DUTY_CYCLE):** Expressed in percentage(%). Clipped to a max value of 100%.
23:17	0h RW/L	**TIME_WINDOW (TIME_WINDOW):** Duration over which duty cycle control will be maintained. (xxYYYY format)
16	0h RO	Reserved.
15	0h RW/L	**PL3_ENABLE (PL3_ENABLE):** 0 => Algorithm disabled, 1 => Algorithm enabled
14:0	0h RW/L	**POWER_LIMIT (POWER_LIMIT):** PL3 or PAppMax power level. A power reading above this will be interpreted as a violation (in increments of 1/8th Watt)

30.15 VR_CURRENT_CONFIG (P_CR_VR_CURRENT_CONFIG)—Offset 47C0h

PUNIT_MSR: VCCGI rail configuration information, including:

- Limitation on the maximum current consumption of the primary compute power plane.
- Configuration of VR light load management, such as PS0, PS1, etc support

Set power state amp thresholds to 0A in order to disable that power state. **MSR_Name / address**: VR_CURRENT_CONFIG / 0x601

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 10h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RSV	**RESERVED_1 (RESERVED_1):** Reserved
62	0h RW	**PS4_ENABLE (PS4_ENABLE):** Enable for PS4 on the VCCGI rail. (current is regulated to ~0A and voltage goes to 0V): • 1 = PSI4 is Enabled • 0 = disable
61:52	0h RW	**VCCGI PS3 Maximum Current (PSI3_THRESHOLD):** Peak current supported by the VR on the VCCGI rail when in single phase mode (PS3). A value of 0 indicates VR support 0A of current in PS3 which will effectively disable PS3 use. This field is defined in 1A units.
51:42	0h RW	**VCCGI PS2 Maximum Current (PSI2_THRESHOLD):** Peak current supported by the VR on the VCCGI rail when in single phase mode (PS2). A value of 0 indicates VR support 0A of current in PS2 which will effectively disable PS2 use. This field is defined in 1A units.
41:32	0h RW	**VCCGI PS1 Maximum Current (PSI1_THRESHOLD):** Peak current supported by the VR on the VCCGI rail when in single phase mode (PS1). A value of 0 indicates VR support 0A of current in PS1 which will effectively disable PS1 use. This field is defined in 1A units.
31	0h RW/L	**LOCK (LOCK):** This bit will lock the CURRENT_LIMIT settings in this register and will also lock this setting. This means that once set to 1b the CURRENT_LIMIT setting and this bit become Read Only until the next Warm Reset.
30:13	0h RSV	**RESERVED_0 (RESERVED_0):** Reserved
12:0	10h RW/L	**VCCGI PS0 Maximum Current (CURRENT_LIMIT):** Peak current supported by the VR on the VCCGI rail when all phases are active. Assumes VR is in the highest power state possible (PS0). Units given in PACKAGE_POWER_SKU_UNIT.CURRENT_UNIT. This field will be locked when the LOCK bit in this register is written to 1b. That lock will be retained until the next reset condition.

30.16 Core Thermal and Power Status (P_CR_THERM_STATUS)—Offset 4828h

PUNIT_MSR: CPU Core thermal and power status register. This register is typically used to inspect the thermal status of a specific core, as well as to discover the sources of core thermal interrupt events. **MSR_Name**: THERM_STATUS **MSR_Addr**: 0x19C

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 8000000h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Valid (VALID):** Valid flag indicating the validity of the current temperature readout. • 0 = Invalid • 1 = Valid
30:27	1h RO	**Resolution (RESOLUTION):** Supported resolution in degrees Celsius. Hardcoded to 1'C.
26:23	0h RSV	**RESERVED1 (RESERVED1):** reserved
22:16	0h RO/V	**Relative Temperature (TEMPERATURE):** Current temperature of the domain, reported as a negative offset from the reference maximum junction temperature. E.g., a reading of 10 with a reference temperature of 90'C implies this domain is currently operating at 80'C. The reference temperature may be discovered in TEMPERATURE_TARGET.REF_TEMP
15:12	0h RSV	**RESERVED0 (RESERVED0):** reserved
11	0h RW/0C/V	**Power Limitation Log (POWER_LIMITATION_LOG):** Sticky log bit that asserts on a rising edge of the Power Limitation Status indicator. This flag remains asserted until software clears it
10	0h RO/V	**Power Limitation Status (POWER_LIMITATION_STATUS):** The power limit status flag asserts at any time that the domain is performance limited due to power constraints. Performance limitation due to power limits is measured as observing the resolved domain level frequency below the software requested frequency, even within the Turbo operating region. The RAPL PL1 and PL2 power limits are the only sources of this power constraint.
9	0h RW/0C/V	**Thermal Threshold 2 Log (THRESHOLD2_LOG):** Sticky log bit that asserts on a rising or falling edge of the thermal threshold 2 status indicator. This flag remains asserted until software clears it
8	0h RO/V	**Thermal Threshold 2 Status (THRESHOLD2_STATUS):** Indicates whether the actual temperature is currently higher than or equal to the value set in Thermal Threshold 2. • 0 = Current temperature is below threshold 2 • 1 = Current temperature at or above threshold 2
7	0h RW/0C/V	**Thermal Threshold 1 Log (THRESHOLD1_LOG):** Sticky log bit that asserts on a rising or falling edge of the thermal threshold 1 status indicator. This flag remains asserted until software clears it
6	0h RO/V	**Thermal Threshold 1 Status (THRESHOLD1_STATUS):** Indicates whether the actual temperature is currently higher than or equal to the value set in Thermal Threshold 1. • 0 = Current temperature is below threshold 1 • 1 = Current temperature at or above threshold 1
5	0h RW/0C/V	**Critical Thermal Event Log (OUT_OF_SPEC_LOG):** Sticky log bit indicating that a Critical Thermal Event was detected since the last time this bit was cleared by software.
4	0h RO/V	**Critical Thermal Event Status (OUT_OF_SPEC_STATUS):** When set, this event indicates that the processor has observed a critical thermal excursion. It is intended as an early warning of thermal runaway in the silicon and shutdown is recommended.
3	0h RW/0C/V	**PROCHOT# Log (PROCHOT_LOG):** Sticky log bit indicating that the PROCHOT# status bit has asserted or deasserted since the last time this bit was cleared by software.
2	0h RO/V	**PROCHOT# Status (PROCHOT_STATUS):** Status bit indicating that the external PROCHOT# pin is currently being asserted. This will only assert when the processor is configured in Bi-directional PROCHOT# mode (when PROCHOT# is configured as an input)
1	0h RW/0C/V	**Thermal Monitor Log (THERMAL_MONITOR_LOG):** Sticky log bit indicating that the core has seen a rising edge on the Thermal Monitor Status bit since the last time this bit was cleared by software.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO/V	**Thermal Monitor Status (THERMAL_MONITOR_STATUS):** This bit indicates that the Thermal Monitor has tripped and is currently thermally throttling this core. If software has programmed a thermal monitor control temperature below the manufacturing default of this processor, this flag will assert when the software programmed constraint has been exceeded.

30.17 Thermal Interrupt Control (P_CR_THERM_INTERRUPT)—Offset 482Ch

PUNIT_MSR: CPU Core thermal interrupt control register. This register is used to configure core or package level notifications of important thermal events. These interrupts may be routed directly to this local core's APIC Thermal LVT or optional modes may be configured that allow the thermal interrupt notification to be routed to a master core for processing (fore more details on locking thermal interrupts, see the MISC_PWR_MGMT MSR definition). **MSR_Name**: THERM_INTERRUPT **MSR_Addr**: 0x19B

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RSV	**RESERVED2 (RESERVED2):** reserved
24	0h RW	**Power Limit Interrupt Enable (SPARE_PWR_LIMIT_ENABLE): NOT USED**. Reserved for Power Limit Notification Enable.
23	0h RW	**Threshold 2 Interrupt Enable (THRESHOLD_2_INT_ENABLE):** Threshold 2 Interrupt Enable is used to enable thermal interrupt notifcation to software upon any rising or falling detection of core temperature crossing the Threshold 2 temperature.
22:16	0h RW	**Thermal Threshold 2 Relative Temperature (THRESHOLD_2_REL_TEMP):** Specifies temperature Threshold 2 value as as a negative offset from the reference maximum junction temperature (defined in TEMPERATURE_TARGET.REF_TEMP). This temperature threshold is compared against the corresponding Thermal Status Relative Temperature reading for this core.
15	0h RW	**Threshold 1 Interrupt Enable (THRESHOLD_1_INT_ENABLE):** Threshold 1 Interrupt Enable is used to enable thermal interrupt notifcation to software upon any rising or falling detection of core temperature crossing the Threshold 1 temperature.
14:8	0h RW	**Thermal Threshold 1 Relative Temperature (THRESHOLD_1_REL_TEMP):** Specifies temperature Threshold 1 value as as a negative offset from the reference maximum junction temperature (defined in TEMPERATURE_TARGET.REF_TEMP). This temperature threshold is compared against the corresponding Thermal Status Relative Temperature reading for this core.
7:5	0h RSV	**RESERVED1 (RESERVED1):** reserved
4	0h RW	**Critical Thermal Event Interrupt Enable (OUT_OF_SPEC_INT_ENABLE):** This bit allows software to enable thermal interrupt delivery upon observation that the processor is operating out of its thermal specification. This interrupt is designed as an early warning of thermal runaway in the silicon and shutdown is recommended.
3	0h RSV	**RESERVED0 (RESERVED0):** reserved

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**PROCHOT# Interrupt Enable (PROCHOT_INT_ENABLE):** This bit allows software to enable a thermal interrupt delivery upon detection of PROCHOT# input assertion or deassertion event.
1	0h RW	**Low Temperature Interrupt Enable (LOW_TEMP_INT_ENABLE):** This bit allows software to enable the generation of an interrupt on the transition from a temperature above or equal to the thermal monitor activation temperature to a temperature below it. This interrupt will be sent on any falling edge of the Thermal Monitor Status bit of this core.
0	0h RW	**High Temperature Interrupt Enable (HIGH_TEMP_INT_ENABLE):** This bit allows software to enable the generation of an interrupt on the transition from a temperature below the thermal monitor activation temperature to a temperature equal to or greater than it. This interrupt will be sent on any rising edge of the Thermal Monitor Status bit of this core.

30.18 C1E Idle Residency (P_CR_PC2_RCNTR)—Offset 4870h

PUNIT_MSR: Time spent in the C1E state. When accessed at the core level via RDMSR, time is measured in the same units as the Maximum Non-Turbo clock frequency. When accessed at the P-unit level, the time units are based on the crystal clock frequency.
MSR_Name: PC2_RCNTR **MSR_Addr**: 0x60D

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:60	0h RO	**RESERVED_0 (RESERVED_0):** Reserved
59:0	0h RW	**DATA (DATA):** Time spent in the C1E state. When accessed at the core level via RDMSR, time is measured in the same units as the Maximum Non-Turbo clock frequency. When accessed at the P-unit level, the time units are based on the crystal clock frequency.

30.19 PC3 Idle Residency Counter (P_CR_PC3_RCNTR)—Offset 4878h

PUNIT_MSR: Currently unused residency counter. When accessed at the core level via RDMSR, time is measured in the same units as the Maximum Non-Turbo clock frequency. When accessed at the P-unit level, the time units are based on the crystal clock frequency.
MSR_Name: PC3_RCNTR **MSR_Addr**: 0x3F8

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:60	0h RO	**RESERVED_0 (RESERVED_0):** Reserved
59:0	0h RW	**COUNTER (COUNTER):** Currently unused residency counter. When accessed at the core level via RDMSR, time is measured in the same units as the Maximum Non-Turbo clock frequency. When accessed at the P-unit level, the time units are based on the crystal clock frequency.

30.20 PCS_Idle_Residency (P_CR_PC6_RCNTR)—Offset 4880h

PUNIT_MSR: Primary Compute System (PCS) idle residency counter. This counter increments at any time that the SOC is in the S0i1 state, with cores, graphics and camera processing system all idle. Display, Camera Input System or other south complex IPs may be active at this time. When accessed at the core level via RDMSR, time is measured in the same units as the Maximum Non-Turbo clock frequency. When accessed at the P-unit level, the time units are based on the crystal clock frequency.
MSR_Name: PC6_RCNTR **MSR_Addr**: 0x3F9

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:60	0h RO	**RESERVED_0 (RESERVED_0):** Reserved
59:0	0h RW	**COUNTER (COUNTER):** Primary Compute System (PCS) idle residency counter. This counter increments at any time that the SOC is in the S0i1 state, with cores, graphics and camera processing system all idle. Display, Camera Input System or other south complex IPs may be active at this time. When accessed at the core level via RDMSR, time is measured in the same units as the Maximum Non-Turbo clock frequency. When accessed at the P-unit level, the time units are based on the crystal clock frequency.

30.21 PCS S0i3 Residency (P_CR_PC7_RCNTR)—Offset 4888h

PUNIT_MSR: Primary Compute System (PCS) S0i3 residency counter. This counter increments at any time that the PCS is ready for S0i3, which includes all compute domains idle (IA cores, graphics and camera) as well as display and camera input system both off. The SOC may remain in this state for extended periods if south complex IPs are not ready to enter S0i3. When accessed at the core level via RDMSR, time is measured in the same units as the Maximum Non-Turbo clock frequency. When accessed at the P-unit level, the time units are based on the crystal clock frequency.
MSR_Name: PC7_RCNTR **MSR_Addr**: 0x3FA

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:60	0h RO	**RESERVED_0 (RESERVED_0):** Reserved
59:0	0h RW	**COUNTER (COUNTER):** Primary Compute System (PCS) S0i3 residency counter. This counter increments at any time that the PCS is ready for S0i3, which includes all compute domains idle (IA cores, graphics and camera) as well as display and camera input system both off. The SOC may remain in this state for extended periods if south complex IPs are not ready to enter S0i3. When accessed at the core level via RDMSR, time is measured in the same units as the Maximum Non-Turbo clock frequency. When accessed at the P-unit level, the time units are based on the crystal clock frequency.

30.22 SOC S0i3 Residency (P_CR_PC10_RCNTR)—Offset 4890h

PUNIT_MSR: SOC S0i3 residency counter. This counter increments at any time that the entire SOC is in an S0i3 state. This register does not distinguish variations on S0i3, it increments for all variations. This residency counter may count some time during entry into and exit from S0i3. When accessed at the core level via RDMSR, time is measured in the same units as the Maximum Non-Turbo clock frequency. When accessed at the P-unit level, the time units are based on the crystal clock frequency. **MSR_Name**: PC10_RCNTR **MSR_Addr**: 0x632

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:60	0h RO	**RESERVED_0 (RESERVED_0):** Reserved
59:0	0h RW	**COUNTER (COUNTER):** SOC S0i3 residency counter. This counter increments at any time that the entire SOC is in an S0i3 state. This register does not distinguish variations on S0i3, it increments for all variations. This residency counter may count some time during entry into and exit from S0i3. When accessed at the core level via RDMSR, time is measured in the same units as the Maximum Non-Turbo clock frequency. When accessed at the P-unit level, the time units are based on the crystal clock frequency.

30.23 LLC_FLUSHED_RCNTR (P_CR_L2_FLUSH_RCNTR)—Offset 4898h

Time spent with LLC flushed. It is given in units compatible to P1 clock frequency (Guaranteed / Maximum Core Non-Turbo Frequency).

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:60	0h RO	**RESERVED_0 (RESERVED_0):** Reserved
59:0	0h RW	**COUNTER (COUNTER):** Counter Value

30.24 PC9 Idle Residency Counter (P_CR_PC9_RCNTR)—Offset 48A0h

PUNIT_MSR: Currently unused residency counter. When accessed at the core level via RDMSR, time is measured in the same units as the Maximum Non-Turbo clock frequency. When accessed at the P-unit level, the time units are based on the crystal clock frequency. **MSR_Name**: PC9_RCNTR **MSR_Addr**: 0x631

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:60	0h RO	**RESERVED_0 (RESERVED_0):** Reserved
59:0	0h RW	**COUNTER (COUNTER):** Currently unused residency counter. When accessed at the core level via RDMSR, time is measured in the same units as the Maximum Non-Turbo clock frequency. When accessed at the P-unit level, the time units are based on the crystal clock frequency.

30.25 C_STATE_LATENCY_CONTROL_0 (P_CR_C_STATE_LATENCY_CONTROL_0)—Offset 48A8h

PUNIT_MSR: This register describes the OS tolerated wake latency for the C3 state. The minimum core C-state is calculated among all IA cores in the die, and power management firmware maps that resolved C-state to the appropriate Interrupt Response Tolerance latency setting. This setting will then affect package level idle state entry decisions such that it is guaranteed that the cores will wake up in less than the tolerated latency.
Registers initialized by Pcode at reset. BIOS cannot update these MSRs prior to PCODE_INIT_DONE, and it is not recommended that BIOS ever overwrite.
MSR_Name: C_STATE_LATENCY_CONTROL_0 **MSR_Addr**: 12'h60A

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RSV	**RESERVED_1 (RESERVED_1):** Reserved.
15	0h RW	**VALID (VALID):** This field qualifies the validity of the Value field in this register. If the valid bit is zero, then it is assumed that software has no constraints on wake latency (i.e., it supports infinity).
14:13	0h RSV	**RESERVED_0 (RESERVED_0):** Reserved.
12:10	0h RW	**MULTIPLIER (MULTIPLIER):** This field indicates the unit of measurement that is defined for the Value field in this register. The units are 2^(5*multiplier) • 000b = 1ns (2^0) • 001b = 32ns (2^5) • 010b = 1024ns (2^10) • 011b = 32.768us (2^15) • 100b = 1.048ms (2^20) • 101b = 33.55ms (2^25)
9:0	0h RW	**VALUE (VALUE):** This scalar is multiplied by the multiplier field to calculate the net latency tolerance. Ex., with a MULTIPLIER of 2 and a VALUE of 20, the net latency tolerance is 20 * (2^10) = 20480ns or 20.48us

30.26 C_STATE_LATENCY_CONTROL_1 (P_CR_C_STATE_LATENCY_CONTROL_1)—Offset 48ACh

PUNIT_MSR: This register describes the OS tolerated wake latency for the C6/C7s state. The minimum core C-state is calculated among all IA cores in the die, and power management firmware maps that resolved C-state to the appropriate Interrupt Response Tolerance latency setting. This setting will then affect package level idle state entry decisions such that it is guaranteed that the cores will wake up in less than the tolerated latency.
Registers initialized by Pcode at reset. BIOS cannot update these MSRs prior to PCODE_INIT_DONE, and it is not recommended that BIOS ever overwrite.
MSR_Name: C_STATE_LATENCY_CONTROL_1 **MSR_Addr**: 0x60B

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RSV	**RESERVED_1 (RESERVED_1):** Reserved
15	0h RW	**VALID (VALID):** This field qualifies the validity of the Value field in this register. If the valid bit is zero, then it is assumed that software has no constraints on wake latency (i.e., it supports infinity).
14:13	0h RSV	**RESERVED_0 (RESERVED_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
12:10	0h RW	**MULTIPLIER (MULTIPLIER):** This field indicates the unit of measurement that is defined for the Value field in this register. The units are 2^(5*multiplier) • 000b = 1ns (2^0) • 001b = 32ns (2^5) • 010b = 1024ns (2^10) • 011b = 32.768us (2^15) • 100b = 1.048ms (2^20) • 101b = 33.55ms (2^25)
9:0	0h RW	**VALUE (VALUE):** This scalar is multiplied by the multiplier field to calculate the net latency tolerance. Ex., with a MULTIPLIER of 2 and a VALUE of 20, the net latency tolerance is 20 * (2^10) = 20480ns or 20.48us

30.27 C_STATE_LATENCY_CONTROL_2 (P_CR_C_STATE_LATENCY_CONTROL_2)—Offset 48B0h

This register describes the OS tolerated wake latency for the C7L state. The minimum core C-state is calculated among all IA cores in the die, and power management firmware maps that resolved C-state to the appropriate Interrupt Response Tolerance latency setting. This setting will then affect package level idle state entry decisions such that it is guaranteed that the cores will wake up in less than the tolerated latency. Registers initialized by Pcode at reset. BIOS cannot update these MSRs prior to PCODE_INIT_DONE, and it is not recommended that BIOS ever overwrite.
MSR_Name: C_STATE_LATENCY_CONTROL_2 **MSR_Addr**: 12'h60C

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RSV	**RESERVED_1 (RESERVED_1):** Reserved.
15	0h RW	**VALID (VALID):** This field qualifies the validity of the Value field in this register. If the valid bit is zero, then it is assumed that software has no constraints on wake latency (i.e., it supports infinity).
14:13	0h RSV	**RESERVED_0 (RESERVED_0):** Reserved.
12:10	0h RW	**MULTIPLIER (MULTIPLIER):** This field indicates the unit of measurement that is defined for the Value field in this register. The units are 2^(5*multiplier) • 000b = 1ns (2^0) • 001b = 32ns (2^5) • 010b = 1024ns (2^10) • 011b = 32.768us (2^15) • 100b = 1.048ms (2^20) • 101b = 33.55ms (2^25)
9:0	0h RW	**VALUE (VALUE):** This scalar is multiplied by the multiplier field to calculate the net latency tolerance. Ex., with a MULTIPLIER of 2 and a VALUE of 20, the net latency tolerance is 20 * (2^10) = 20480ns or 20.48us

30.28 C_STATE_LATENCY_CONTROL_3 (P_CR_C_STATE_LATENCY_CONTROL_3)—Offset 48B4h

PUNIT_MSR: This register describes the OS tolerated wake latency for the C8 state. The minimum core C-state is calculated among all IA cores in the die, and power management firmware maps that resolved C-state to the appropriate Interrupt Response Tolerance latency setting. This setting will then affect package level idle state entry decisions such that it is guaranteed that the cores will wake up in less than the tolerated latency.
Registers initialized by Pcode at reset. BIOS cannot update these MSRs prior to PCODE_INIT_DONE, and it is not recommended that BIOS ever overwrite.
MSR_Name: C_STATE_LATENCY_CONTROL_3 **MSR_Addr**: 0x633

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RSV	**RESERVED_1 (RESERVED_1):** Reserved
15	0h RW	**VALID (VALID):** This field qualifies the validity of the Value field in this register. If the valid bit is zero, then it is assumed that software has no constraints on wake latency (i.e., it supports infinity).
14:13	0h RSV	**RESERVED_0 (RESERVED_0):** Reserved
12:10	0h RW	**MULTIPLIER (MULTIPLIER):** This field indicates the unit of measurement that is defined for the Value field in this register. The units are 2^(5*multiplier) • 000b = 1ns (2^0) • 001b = 32ns (2^5) • 010b = 1024ns (2^10) • 011b = 32.768us (2^15) • 100b = 1.048ms (2^20) • 101b = 33.55ms (2^25)
9:0	0h RW	**VALUE (VALUE):** This scalar is multiplied by the multiplier field to calculate the net latency tolerance. Ex., with a MULTIPLIER of 2 and a VALUE of 20, the net latency tolerance is 20 * (2^10) = 20480ns or 20.48us

30.29 C_STATE_LATENCY_CONTROL_4 (P_CR_C_STATE_LATENCY_CONTROL_4)—Offset 48B8h

PUNIT_MSR: This register describes the OS tolerated wake latency for the C9 state. The minimum core C-state is calculated among all IA cores in the die, and power management firmware maps that resolved C-state to the appropriate Interrupt Response Tolerance latency setting. This setting will then affect package level idle state entry decisions such that it is guaranteed that the cores will wake up in less than the tolerated latency.
Registers initialized by Pcode at reset. BIOS cannot update these MSRs prior to PCODE_INIT_DONE, and it is not recommended that BIOS ever overwrite.
MSR_Name: C_STATE_LATENCY_CONTROL_4 **MSR_Addr**: 0x634

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RSV	**RESERVED_1 (RESERVED_1):** Reserved
15	0h RW	**VALID (VALID):** This field qualifies the validity of the Value field in this register. If the valid bit is zero, then it is assumed that software has no constraints on wake latency (i.e., it supports infinity).
14:13	0h RSV	**RESERVED_0 (RESERVED_0):** Reserved
12:10	0h RW	**MULTIPLIER (MULTIPLIER):** This field indicates the unit of measurement that is defined for the Value field in this register. The units are 2^(5*multiplier) • 000b = 1ns (2^0) • 001b = 32ns (2^5) • 010b = 1024ns (2^10) • 011b = 32.768us (2^15) • 100b = 1.048ms (2^20) • 101b = 33.55ms (2^25)
9:0	0h RW	**VALUE (VALUE):** This scalar is multiplied by the multiplier field to calculate the net latency tolerance. Ex., with a MULTIPLIER of 2 and a VALUE of 20, the net latency tolerance is 20 * (2^10) = 20480ns or 20.48us

30.30 C_STATE_LATENCY_CONTROL_5 (P_CR_C_STATE_LATENCY_CONTROL_5)—Offset 48BCh

PUNIT_MSR: This register describes the OS tolerated wake latency for the C10 state. The minimum core C-state is calculated among all IA cores in the die, and power management firmware maps that resolved C-state to the appropriate Interrupt Response Tolerance latency setting. This setting will then affect package level idle state entry decisions such that it is guaranteed that the cores will wake up in less than the tolerated latency.
Registers initialized by Pcode at reset. BIOS cannot update these MSRs prior to PCODE_INIT_DONE, and it is not recommended that BIOS ever overwrite.
MSR_Name: C_STATE_LATENCY_CONTROL_5 **MSR_Addr**: 12'h635

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RSV	**RESERVED_1 (RESERVED_1):** Reserved
15	0h RW	**VALID (VALID):** This field qualifies the validity of the Value field in this register. If the valid bit is zero, then it is assumed that software has no constraints on wake latency (i.e., it supports infinity).

Bit Range	Default & Access	Field Name (ID): Description
14:13	0h RSV	**RESERVED_0 (RESERVED_0):** Reserved
12:10	0h RW	**MULTIPLIER (MULTIPLIER):** This field indicates the unit of measurement that is defined for the Value field in this register. The units are 2^(5*multiplier) • 000b = 1ns (2^0) • 001b = 32ns (2^5) • 010b = 1024ns (2^10) • 011b = 32.768us (2^15) • 100b = 1.048ms (2^20) • 101b = 33.55ms (2^25)
9:0	0h RW	**VALUE (VALUE):** This scalar is multiplied by the multiplier field to calculate the net latency tolerance. Ex., with a MULTIPLIER of 2 and a VALUE of 20, the net latency tolerance is 20 * (2^10) = 20480ns or 20.48us

30.31 Package Thermal and Power Status (P_CR_PACKAGE_THERM_STATUS)—Offset 49F0h

PUNIT_MSR: Package level thermal and power status register. This register is typically used to inspect the thermal status of the entire package, as well as to discover the sources of package level thermal interrupt events. **MSR_Name**: THERM_STATUS
MSR_Addr: 0x19C

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 8000000h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/V	**Valid (VALID):** Valid flag indicating the validity of the current temperature readout. • 0 = Invalid • 1 = Valid
30:27	1h RO	**Resolution (RESOLUTION):** Supported resolution in degrees Celsius. Hardcoded to 1'C.
26:23	0h RSV	**RESERVED_1 (RESERVED_1):** Reserved
22:16	0h RO/V	**Relative Temperature (TEMPERATURE):** Current maximum temperature across all sensors in the package, reported as a negative offset from the reference maximum junction temperature. E.g., a reading of 10 with a reference temperature of 90'C implies this domain is currently operating at 80'C. The reference temperature may be discovered in TEMPERATURE_TARGET.REF_TEMP
15:12	0h RSV	**RESERVED_0 (RESERVED_0):** Reserved
11	0h RW/0C/V	**Power Limitation Log (POWER_LIMITATION_LOG):** Sticky log bit that asserts on a rising edge of the Power Limitation Status indicator. This flag remains asserted until software clears it
10	0h RO/V	**Power Limitation Status (POWER_LIMITATION_STATUS):** The power limit status flag asserts at any time that the domain is performance limited due to power constraints. Performance limitation due to power limits is measured as observing the resolved domain level frequency below the software requested frequency, even within the Turbo operating region. The RAPL PL1 and PL2 power limits are the only sources of this power constraint.

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW/0C/V	**Thermal Threshold 2 Log (THRESHOLD2_LOG):** Sticky log bit that asserts on a rising or falling edge of the thermal threshold 2 status indicator. This flag remains asserted until software clears it
8	0h RO/V	**Thermal Threshold 2 Status (THRESHOLD2_STATUS):** Indicates whether the actual temperature is currently higher than or equal to the value set in Thermal Threshold 2. • 0 = Current temperature is below threshold 2 • 1 = Current temperature at or above threshold 2
7	0h RW/0C/V	**Thermal Threshold 1 Log (THRESHOLD1_LOG):** Sticky log bit that asserts on a rising or falling edge of the thermal threshold 1 status indicator. This flag remains asserted until software clears it
6	0h RO/V	**Thermal Threshold 1 Status (THRESHOLD1_STATUS):** Indicates whether the actual temperature is currently higher than or equal to the value set in Thermal Threshold 1. • 0 = Current temperature is below threshold 1 • 1 = Current temperature at or above threshold 1
5	0h RW/0C/V	**Critical Thermal Event Log (OUT_OF_SPEC_LOG):** Sticky log bit indicating that a Critical Thermal Event was detected since the last time this bit was cleared by software.
4	0h RO/V	**Critical Thermal Event Status (OUT_OF_SPEC_STATUS):** When set, this event indicates that the processor has observed a critical thermal excursion. It is intended as an early warning of thermal runaway in the silicon and shutdown is recommended.
3	0h RW/0C/V	**PROCHOT# Log (PROCHOT_LOG):** Sticky log bit indicating that the PROCHOT# status bit has asserted or deasserted since the last time this bit was cleared by software.
2	0h RO/V	**PROCHOT# Status (PROCHOT_STATUS):** Status bit indicating that the external PROCHOT# pin is currently being asserted. This will only assert when the processor is configured in Bi-directional PROCHOT# mode (when PROCHOT# is configured as an input)
1	0h RW/0C/V	**Thermal Monitor Log (THERMAL_MONITOR_LOG):** Sticky log bit indicating that the package has seen a rising edge on the Thermal Monitor Status bit since the last time this bit was cleared by software.
0	0h RO/V	**Thermal Monitor Status (THERMAL_MONITOR_STATUS):** This bit indicates that the Thermal Monitor has tripped and is currently thermally throttling at least one domain in the package. If software has programmed a thermal monitor control temperature below the manufacturing default of this processor, this flag will assert when the software programmed constraint has been exceeded.

30.32 Thermal Interrupt Control (P_CR_PACKAGE_THERM_INTERRUPT)—Offset 49F4h

PUNIT_MSR: Package thermal interrupt control register. This register is used to configure package level notifications of important thermal events. These interrupts are routed directly to core APIC Thermal LVT's and are handled locally by the core that has unmasked the LVT. **MSR_Name**: THERM_INTERRUPT **MSR_Addr**: 0x19B THIS REGISTER IS DUPLICATED IN THE PCU IO SPACE, XML CHANGES MUST BE MADE IN BOTH PLACES

Access Method

Type: MSG Register
(Size: 32 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RSV	**RESERVED_2 (RESERVED_2):** Reserved
24	0h RW	**Power Limit Interrupt Enable (POWER_INT_ENABLE):** When this bit is set a thermal interrupt will be sent upon detection of a rising edge assertion of the Power Limit Status bit in the Package Thermal and Power Status register.
23	0h RW	**Threshold 2 Interrupt Enable (THRESHOLD_2_INT_ENABLE):** Threshold 2 Interrupt Enable is used to enable thermal interrupt notifcation to software upon any rising or falling detection of the virtual package max temperature crossing the Threshold 2 temperature.
22:16	0h RW	**Thermal Threshold 2 Relative Temperature (THRESHOLD_2_REL_TEMP):** Specifies temperature Threshold 2 value as as a negative offset from the reference maximum junction temperature (defined in TEMPERATURE_TARGET.REF_TEMP). This temperature threshold is compared against the corresponding Thermal Status Relative Temperature reading for the virtual maximum package temperature.
15	0h RW	**Threshold 1 Interrupt Enable (THRESHOLD_1_INT_ENABLE):** Threshold 1 Interrupt Enable is used to enable thermal interrupt notifcation to software upon any rising or falling detection of the virtual package max temperature crossing the Threshold 1 temperature.
14:8	0h RW	**Thermal Threshold 1 Relative Temperature (THRESHOLD_1_REL_TEMP):** Specifies temperature Threshold 1 value as as a negative offset from the reference maximum junction temperature (defined in TEMPERATURE_TARGET.REF_TEMP). This temperature threshold is compared against the corresponding Thermal Status Relative Temperature reading for the virtual maximum package temperature.
7:5	0h RSV	**RESERVED_1 (RESERVED_1):** Reserved
4	0h RW	**Critical Thermal Event Interrupt Enable (OUT_OF_SPEC_INT_ENABLE):** This bit allows software to enable thermal interrupt delivery upon observation that the processor is operating out of its thermal specification. This interrupt is designed as an early warning of thermal runaway in the silicon and shutdown is recommended.
3	0h RSV	**RESERVED_0 (RESERVED_0):** Reserved
2	0h RW	**PROCHOT# Interrupt Enable (PROCHOT_INT_ENABLE):** This bit allows software to enable a thermal interrupt delivery upon detection of PROCHOT# input assertion or deassertion event.
1	0h RW	**Low Temperature Interrupt Enable (LOW_TEMP_INT_ENABLE):** This bit allows software to enable the generation of an interrupt on the transition from a temperature above or equal to the thermal monitor activation temperature to a temperature below it. This interrupt will be sent on any falling edge of the Thermal Monitor Status bit of the virtual maximum package temperature.
0	0h RW	**High Temperature Interrupt Enable (HIGH_TEMP_INT_ENABLE):** This bit allows software to enable the generation of an interrupt on the transition from a temperature below the thermal monitor activation temperature to a temperature equal to or greater than it. This interrupt will be sent on any rising edge of the Thermal Monitor Status bit of this virtual maximum temperature.

30.33 IA32_MC6_STATUS (A_CR_MCI_STATUS)—Offset 4F80h

Machine check control.

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW/V	**Valid (VALID):** MCI_Status Valid (VAL): When set, indicates the information on the IA32_MCI_Status register is valid. Software is responsible for clearing the VAL flag.
62	0h RW/V	**Overflow (OVFLW):** Error Overflow (OVER): Indicates a second error was received while the MCI_Status Valid bit as set.
61	0h RW/V	**Uncorrected Error (UC):** Indicates that an uncorrected error was received by the A-Unit.
60	0h RW/V	**Error Report Enabled (EN):** Error reporting has been enabled by the MCI_CTL register.
59	0h RW/V	**MISC Valid (MISCV):** MISCV (mci_misc is valid).
58	0h RW/V	**Address Valid (ADDRV):** ADDRV (mci_addr is valid).
57	0h RW/V	**Process Context Corrupted (PCC):** When set, indicates the state of the processor might have been corrupted. When clear, indicates the error did not affect the processor state.
56	0h RW/V	**Signal (S):** S (signaling an uncorr recoverable (UCR) error).
55	0h RW/V	**Recovery Action Required (AR):** AR (recovery action required for UCR error).
54:53	0h RW/V	**Threshold Status (THRSH):** Threshold-based error status.
52:38	0h RW/V	**Corectable Error Count (CERR_CNT)**
37:32	0h RW/V	**Other (OTHER)**
31:16	0h RW/V	**Model Specific Error Code (MODEL_SPECIFIC_ERR_CODE)**
15:0	0h RW/V	**MCA Error Code (MCA_ERR_CODE)**

30.34 MCA Control Register (A_CR_MCI_CTL)—Offset 4F88h

Controls which errors will signal an MCA. 0 - do not signal. 1 - signal.

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:32	0h RO	Reserved.
31	0h RO	**reserved (undefined31):** reserved for future use
30	0h RO	**reserved (undefined30):** reserved for future use

Bit Range	Default & Access	Field Name (ID): Description
29	0h RO	**reserved (undefined29):** reserved for future use
28	0h RO	**reserved (undefined28):** reserved for future use
27	0h RO	**reserved (undefined27):** reserved for future use
26	0h RO	**reserved (undefined26):** reserved for future use
25	0h RO	**reserved (undefined25):** reserved for future use
24	0h RO	**reserved (undefined24):** reserved for future use
23	0h RO	**reserved (undefined23):** reserved for future use
22	0h RO	**reserved (undefined22):** reserved for future use
21	0h RO	**reserved (undefined21):** reserved for future use
20	0h RO	**reserved (undefined20):** reserved for future use
19	0h RO	**reserved (undefined19):** reserved for future use
18	0h RO	**reserved (undefined18):** reserved for future use
17	0h RO	**reserved (undefined17):** reserved for future use
16	0h RO	**reserved (undefined16):** reserved for future use
15	0h RO	**reserved (undefined15):** reserved for future use
14	0h RO	**reserved (undefined14):** reserved for future use
13	0h RO	**reserved (undefined13):** reserved for future use
12	0h RO	**reserved (undefined12):** reserved for future use
11	0h RO	**reserved (undefined11):** reserved for future use
10	0h RW	**msi rsvd set (msi_rsvd_set):** An MSI was received with reserved bits set
9	0h RW	**gpa overflow (gpa_overflow):** A transaction was received with a guest physical address that was too large
8	0h RW	**illegal msi (illegal_msi):** A malformed/illegal MSI was received in the upstream direction
7	0h RW	**at translated illegal device (at_translated_illegal_device):** A device that is not support to set the AT bit set it to an illegal value
6	0h RO	**reserved (undefined5):** reserved for future use

Bit Range	Default & Access	Field Name (ID): Description
5	0h RW	**bad sai cmpl (bad_sai_cmpl):** An incorrect/illegal sai was received with an upstream completion transaction.
4	0h RW	**received lk cmpl (received_lk_cmpl):** Received a CmplLck completion from iosf.
3	0h RW	**bad sai nonposted (bad_sai_nonposted):** An incorrect/illegal sai was received with an upstream non-posted transaction.
2	0h RW	**illegal nonposted opcode (illegal_nonposted_opcode):** Illegal/Unsupported non-posted opcode received from iosf.
1	0h RW	**bad sai posted (bad_sai_posted):** An incorrect/illegal sai was received with an upstream posted transaction.
0	0h RW	**illegal posted opcode (illegal_posted_opcode):** Illegal/Unsupported posted opcode received from iosf.

30.35 IA32_MC6_CTL2 (A_CR_MCI_CTL2)—Offset 4F90h

Machine Check Control 2.

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:31	0h RO	Reserved.
30	0h RW	**Enable (ENABLE):** MCI_CTL2 enable for CMCI.
29:15	0h RO	Reserved.
14:0	0h RW	**Threshold (THRESHOLD):** MCI_CTL2 threshold value for CMCI.

30.36 IA32_MC6_ADDR (A_CR_MCI_ADDR)—Offset 4F98h

Machine check address.

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:0	0h RW/V	**Address (ADDRESS):** MCI_ADDR: Address associated with the MCI_STS register if addrv==1. This is the GPA.

30.37 IA32_MC6_MISC (A_CR_MCI_MISC)—Offset 4FC0h

The MCi Misc MSR contains additional microarchitecture specific information describing the machine check error if the MISCV flag in the MCi_STATUS register is set.

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:48	0h RW/V	**BDF (BDF):** IOSF rqid of request
47:40	0h RW/V	**SAI (SAI):** IOSF SAI (8-bit) of request
39:9	0h RO	Reserved.
8:6	0h RW/V	**Address Mode (ADDR_MODE):** Address Mode
5:0	0h RO	Reserved.

30.38 IA32_MC4_CTL (B_CR_MCI_CTL)—Offset 4B20h

This register specifies which errors are enabled for reporting. When set to 0, error is still logged but not signalled. When set to 1, error is logged and signalled.

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:10	0h RO	**Reserved (RESERVED_0):** Reserved.
9	0h RW	**MC Uncorrected Read Data Error (MC_RD_DATA_UNC):** CRC error on DRAM read data
8	0h RW	**BRAM Read Parity Error (BRAM_RD_PAR):** Userver only feature.

Bit Range	Default & Access	Field Name (ID): Description
7	0h RW	**BRAM Write Parity Error (BRAM_WR_PAR):** Userver only feature.
6	0h RW	**XUCode Software Initiated Error (XUCODE_ERR)**
5	0h RW	**IA Hit to Graphics Stolen Memory (IA_HIT_GSM)**
4	0h RW	**ClFlush to MMIO (MMIO_CLFLUSH)**
3	0h RW	**WBMTo Access to MMIO (MMIO_WBMTOIE)**
2	0h RW	**MMIO Access HITM (MMIO_HITM):** MMIO Access for HIT Modified Snoop Response.
1	0h RW	**Pondicherry Internal Interface Access to MMIO (PII_2_MMIO):** NonIDI access to MMIO.
0	0h RW	**Unsupported IDI Opcode (UNS_IDI_OP)**

30.39 IA32_MC4_STATUS (B_CR_MCI_STATUS)—Offset 4B28h

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW/V	**MCI Status Valid (MCI_STATUS_VALID):** VAL: When set, indicates the information in the rest of this register is valid. Software is responsible for clearing the VAL flag.
62	0h RW/V	**Error Overflow (ERR_OVERFLOW):** Indicates a second error was received while the MCI Status Valid bit was set. The B-Unit supports the Machine Check Overwrite rules for a Machine Check Overflow. For the purpose of B-Unit error reporting, corrected errors or SERRORs sent from the Dunit will be overwritten by any new errors received. The first uncorrected error received will not be overwritten by any new errors received.
61	0h RW/V	**Uncorrected Error (UNCORRECTED_ERR):** UC: Indicates that an uncorrected error was received by the B-Unit. The source of the uncorrected error can be any of those as indicated in the MCI_CTL register.
60	0h RW/V	**Error Reporting Enabled (ERR_ENABLED):** Error reporting has been enabled by the MCI_CTL register.
59	0h RW/V	**MCI_MISC Register Valid (MCI_STATUS_MISCV):** This field will be 0, except for XuCode written errors, which set this field to 1.
58	0h RW/V	**MCI_ADDR Register Valid (MCI_STATUS_ADDRV):** ADDRV: When set, indicates that the value stored in the MCI_ADDR register is valid.
57	0h RW/V	**Processor Context Corrupted (PROCESSOR_CONTEXT_CORRUPTED):** PCC: When set, indicates the state of the processor might have been corrupted. When clear, indicates the error did not affect the processor state.
56:53	0h RO	**Reserved (RESERVED_0):** Reserved. These bits correspond to S AR and Corrected Error Status indicator. They are all readonly and always return 0s

Bit Range	Default & Access	Field Name (ID): Description
52	0h RW/V	**Corrected Error Count Overflow (CORRECTED_ERR_COUNT_OVERFLOW):** Sets if the corrected error count is saturated to 3FFh and another corrected error occurs.
51:38	0h RW/V	**Corrected Error Count (CORRECTED_ERR_COUNT):** The number of corrected errors that have been received from the D-Unit.
37	0h RO	**Reserved (RESERVED_1):** This bit is RO and will always return 0
36:32	0h RW/V	**Agent ID (AGENT_ID):** Agent ID of request that had an ECC or BRAM error.
31:16	0h RW/V	**Model Specific Error Code (MODEL_SPECIFIC_ERR_CODE):** Model Specific Error Code. 0x0000 - Unsupported IDI opcode, 0x0001 - NonIDI access to MMIO, 0x0002 - MMIO Access HITM, 0x0003 - WBMTo* access to MMIO, 0x0004 - ClFlush to MMIO, 0x0005 - IA hit to graphics stolen memory, 0x0006 - XUCode Software initiated error, 0x0007 - BRAM write parity error, 0x0008 - BRAM read parity error, 0x0009 - DRAM uncorrectable ecrc error, 0x000A - DRAM correctable ecrc error
15:0	0h RW/V	**MCA Error Code (MCA_ERR_CODE):** • 0x0005: Parity Error. • 0x009X: Memory read error (where X is channel ID). • 0x408: All other errors.

30.40 IA32_MC4_ADDR (B_CR_MCI_ADDR)—Offset 4B30h

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:40	0h RO	**Reserved (RESERVED_0):** Reserved.
39:0	0h RW/V	**Address MCI_ADDR (MCA_ADDRESS):** When ADDRV is set in the MCI_STATUS register, the memory or system address is stored in this register.

30.41 IA32_MC4_MISC (B_CR_MCI_MISC)—Offset 4B38h

The MCi Misc MSR contains additional microarchitecture specific information describing the machine check error if the MISCV flag in the MCi_STATUS register is set.

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:9	0h RW/V	**Misc Info (MISC):** Miscellaneous information. See register description.

Bit Range	Default & Access	Field Name (ID): Description
8:6	0h RW/V	**Address Mode (ADDR_MODE):** • 000: Segment Offset • 001: Linear Address • 010: Physical Address • 011: Memory Address • 100-110: Reserved • 111: Generic
5:0	0h RW/V	**Recoverable Address LSB (RECOV_ADDR_LSB)**

30.42 IA32_MC4_CTL2 (B_CR_MCI_CTL2)—Offset 4B40h

This register is responsible for enabling and controlling the CMCI mechanism.

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:31	0h RO	**Reserved (RESERVED_1):** Reserved.
30	0h RW	**Corrected Error Interrupt Enable (CORRECTED_ERROR_INTERRUPT_ENABLE):** This bit is set by software to enable the generation of corrected error interrupts.
29:15	0h RO	**Reserved (RESERVED_0):** Reserved.
14:0	0h RW	**Corrected Error Count Threshold (CORRECTED_ERROR_COUNT_THRESHOLD):** Specifies the value to use for the corrected error threshold.

30.43 B-Unit Slice and Channel Hash Function (B_CR_SLICE_CHANNEL_HASH)—Offset 4C58h

A-Unit also has a copy of this register in A_CR_SLICE_CHANNEL_HASH. BIOS MUST explicitly program each register separately and to identical values.

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 3C00000000h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW	**Lock WRMSR (LOCK):** When set, microcode will disallow writes to the this register via the WRMSR path. Intended usage is for BIOS to set the LOCK when it updates the CR, and Ucode to check the LOCK bit to decide whether the WRMSR to this CR must be allowed. B-Unit implements only storage for this bit. No hardware exists to implement hardware locking.
62:52	0h RO	**Reserved (RESERVED_3):** Reserved.
51:38	0h RW	**Channel Hash Mask (CH_HASH_MASK):** When both PMI channels in a slice are enabled, this field specifies the Channel Hash Mask to be applied on Addr[19:6] postremap DRAM address of the request to compute which PMI channel a request must be routed to. Relevant only when HVM mode is disabled, and only for requests that do not fall under the MOT region. B-Unit will override the programmed value to include the Channel Selector bit. Additionally, B-Unit will remove the Slice Selector bit. See INTERLEAVE_MODE field. Note that HVM mode and MOT regions have special hash requirements and hence they do not use the CH_HASH_MASK.
37:36	3h RW	**Channel Enabled for Slice 1 (SYM_SLICE1_CHANNEL_ENABLED):** Specifies which channels in Slice1 are enabled for hosting the symmetric region of the DRAM address space. If SLICE_1_DISABLED is set to disable mapping of DRAM address space to Slice 1, this field has no effect and symmetric region is considered to be not mapped to Slice1. If CH_1_DISABLED is set, Channel 1 in Slice 1 is not enabled for symmetric DRAM address space regardless of the setting of this register. DRAM address space is divided into a symmetric regions, asymmetric regions that are partially interleaved across a subset of channels (2-way) or asymmetric regions that are non-interleaved and mapped only to a single channel. This field specifies which channels in Slice 1 are enabled for interleaving the symmetric region. SYM_SLICE0_CHANNEL_ENABLED defines which channels in Slice 0 are enabled for interleaving the symmetric region. The mapping of partial (2-way) interleaved and non-interleaved regions are specified in other registers. For symmetric region, only selected configurations are supported. Across both slices, only a total of 1 channel, 2 channels or 4 channels must be enabled for interleaving the symmetric region. The total of 1, 2 or 4 channels can be made up of any combination of channel enables across both slices. B-unit does not support a 3-way interleaving of the symmetric region across 3 enabled channels in the two slices combined.
35:34	3h RW	**Channel Enabled for Slice 0 (SYM_SLICE0_CHANNEL_ENABLED):** Specifies which channels in Slice0 are enabled for hosting the symmetric region of the DRAM address space. If SLICE_0_MEM_DISABLED is set to disable mapping of DRAM address space to Slice 0, this field has no effect and symmetric region is considered to be not mapped to Slice0. If CH_1_DISABLED is set, Channel 1 in Slice 0 is not enabled for symmetric DRAM address space regardless of the setting of this register. DRAM address space is divided into a symmetric regions, asymmetric regions that are partially interleaved across a subset of channels (2-way) or asymmetric regions that are non-interleaved and mapped only to a single channel. This field specifies which channels in Slice 0 are enabled for interleaving the symmetric region. SYM_SLICE1_CHANNEL_ENABLED defines which channels in Slice 1 are enabled for interleaving the symmetric region. The mapping of partial (2-way) interleaved and non-interleaved regions are specified in other registers. For symmetric region, only selected configurations are supported. Across both slices, only a total of 1 channel, 2 channels or 4 channels must be enabled for interleaving the symmetric region. The total of 1, 2 or 4 channels can be made up of any combination of channel enables across both slices. B-unit does not support a 3-way interleaving of the symmetric region across 3 enabled channels in the two slices combined.
33	0h RO	**Reserved (RESERVED_2):** Reserved.
32	0h RO	**Channel 1 Disabled (CH_1_DISABLED):** B-Unit is divided into two slices, and each slice has two PMI channels to the internal memory subsystem -- for a total of four channels. With this bit asserted, channel 1 in both slices is disabled. Thus, no memory addresses are mapped to the slice's ch 1. All requests are sent to the slice's channel 0. This bit can be set for SoC Usage of this bit is depricated, Hardware assumes this bit to be always cleared
31	0h RW	**Enable PMI Dual Data Mode (ENABLE_PMI_DUAL_DATA_MODE):** When set to 1, Single Command Interface and Dual Data Interface for Reads and Writes
30:20	0h RO	**Reserved (RESERVED_1):** Reserved.

Bit Range	Default & Access	Field Name (ID): Description
19:6	0h RW	**Slice Hash Mask (SLICE_HASH_MASK):** When both slices are enabled, this field specifies that the Slice Hash Mask is to be applied on Addr[19:6] physical address of the request, to compute which slice a request must be routed to. Relevant only when HVM mode is disabled, and only for physical addresses that do not fall under the Asymmetric Memory Region and the MOT region. B-Unit will override the programmed value to include the Slice Selector bit. Additionally, B-Unit will remove the Channel Selector bit. See INTERLEAVE_MODE field. Note that HVM mode non-address IDI requests asymmetric memory region and MOT regions have special hash requirements and hence they do not use the SLICE_HASH_MASK.
5	0h RO	**Reserved (RESERVED_0):** Reserved.
4	0h RW	**Slice 0 Memory Disabled (SLICE_0_MEM_DISABLED):** Slice 0 is disabled for memory accesses; no memory address mapped to Slice 0 and all memory requests sent to Slice 1.
3:2	0h RW	**Interleave Mode (INTERLEAVE_MODE):** Default interleave mode that specifies how the Slice Selector and Channel Selector bits are to be determined. Relevant only when HVM mode is disabled, and only for system memory addresses that do not fall under the MOT region or the Asymmetric memory region in the System Address Map. Legal encodings are 0h, 1h and 2h. An encoding of 3h is treated as if it was 2h. When both slices and all four PMI channels are enabled: • 0h: Default Slice Selector is Addr[10] and Default Channel Selector is Addr[11] • 1h: Default Slice Selector is Addr[11] and Default Channel Selector is Addr[12] • 2h: Default Slice Selector is Addr[12] and Default Channel Selector is Addr[13] When both slices are enabled, but only one channel in each slice is enabled: • 0h: Default Slice Selector is Addr[10] • 1h: Default Slice Selector is Addr[11] • 2h: Default Slice Selector is Addr[12] When only SLICE0 is enabled, and both channels on that slice are enabled: • 0h: Default Channel Selector is Addr[10] • 1h: Default Channel Selector is Addr[11] • 2h: Default Channel Selector is Addr[12] When SLICE0 and only one channel in that slice is enabled this field is not relevant. B-Unit overrides the setting of the SLICE_HASH_MASK to always include the Slice Selector bit. Similarly, B-Unit overrides the setting of the CH_HASH_MASK to always include the Channel Selector bit.
1	0h RW	**HVM Mode Enable (HVM_MODE):** • 0: HVM mode is disabled. • 1: HVM mode is enabled. When HVM mode is enabled, Slice Hash and Channel Hash are done as follows: Both slices and all four PMI channels enabled: Slice Hash is Request Physical Addr[29] and Channel Hash is PostRemap Addr[30]. Both slices enabled, but only one PMI channel in each slice enabled: Slice Hash is Request Physical Addr[29]. Only one SLICE0 enabled, but both PMI channels in SLICE0 enabled: Channel Hash is PostRemap Addr[29]. When HVM_MODE is enabled, TOLUD must be set at 2GB.
0	0h RW	**Slice 1 Disabled (SLICE_1_DISABLED):** Slice 1 is disabled -- no memory address is mapped to Slice 1. All requests are sent to Slice 0.

30.44 SMM_BLOCKED (T_CR_SMM_BLOCKED)—Offset 5100h

This CSR contains bit per logical processor indicating whether the logical processor is in state where SMIs are blocked and hence will not be able to service SMI. SMI can be serviced after the logical processor exits the state in which SMIs are blocked

Access Method

Type: MSG Register
(Size: 64 bits)

Default: FFFFh

Bit Range	Default & Access	Field Name (ID): Description
63:16	0h RSV	**RESERVED_0 (RESERVED_0):** Reserved
15:0	FFFFh RW/V	**LOG_PROC (LOG_PROC):** This bit will be set when the logical processor is in the following states Wait For SIPI SENTER Sleep VMX Abort Error Shutdown Machine check in WFS state This bit will only exist for logical processors that exist tied to number of logical processors in the system. It is undefined for processors that dont exist.

30.45 SMM_DELAYED (T_CR_SMM_DELAYED)—Offset 5108h

This CSR contains bit per logical processor indicating whether the logical processor is in the middle of long flow and hence will delay servicing of SMI. SMI will be serviced after the long flow completes execution.

Access Method

Type: MSG Register
(Size: 64 bits)

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:16	0h RSV	**RESERVED_0 (RESERVED_0):** Reserved
15:0	0h RW/V	**LOG_PROC (LOG_PROC):** This bit will be set at the start of the flows listed below and cleared at the end of the flows Patch load C6 entry C6 exit WBINVD Ratio Change/ Throttle/S1 This bit will only exist for logical processors that exist tied to number of logical processors in the system. It is undefined for processors that dont exist.

§ §

31 MODPHY Registers

31.1 CLK_SRC_SELECT Register (CLK_SRC_SELECT)—Offset 0h

Clock Source Select Register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**RESERVED3:** Reserved
5	0h RW	**TX_TIMING_CFG_SRC:** Selects the source for timing parameter configuration for DFE 0-Controller, 1-Local Register
4	0h RW	**RESERVED2:** TXDDR_CLK_SRC, Reserved MUST BE 1'b0.
3	0h RW	**TXBYTE_CLK_SRC:** txByte clock source 0=txbyteclock external 1= derived from TxddrclkIhs For Functional mode this bit MUST be set to 1'b0
2:1	0h RW	**RESERVED1**
0	0h RW	**RESERVED0:** Reserved

31.2 TXESC_CLK_DIV1 Register (TXESC_CLK_DIV1)—Offset 4h

txesc div1 register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 80h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RW	**RESERVED0:** Reserved
9:0	80h RW	**TXESC_CLKDIV1:** txesc clock divider 1 ratio(one hot encoding)

31.3 TXESC_CLK_DIV2 Register (TXESC_CLK_DIV2)—Offset 8h

txesc div2 register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 10h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RW	**RESERVED0:** Reserved
9:0	10h RW	**TXESC_CLKDIV2:** txesc clock divider 2 ratio(one hot encoding)

31.4 RCOMP_CLK_DIV1 Register (RCOMP_CLK_DIV1)—Offset Ch

rcomp div1 register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 10h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RW	**RESERVED0:** Reserved
9:0	10h RW	**RCOMP_CLKDIV1:** rcomp clock divider 1 ratio(one hot encoding). By default this is based on i_phyclk (100MHz)

31.5 RCOMP_CLK_DIV2 Register (RCOMP_CLK_DIV2)—Offset 10h

rcomp div2 register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
31:10	0h RW	**RESERVED0:** Reserved
9:0	1h RW	**RCOMP_CLKDIV2:** rcomp clock divider 1 ratio(one hot encoding). By default this is based on i_phyclk (100MHz)

31.6 TOP_CFG_REG Register (TOP_CFG_REG)—Offset 14h

Top Config Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**RESERVED1**
30:3	0h RW	**RESERVED0:** Reserved
2	0h RW	**IP_BI_MODE:** IP_BI_Mode will be used to isolate IP burn-in mode (i.e. near-end analog loopback for MPHY) from SOC full-chip scan mode. IP_BI_Mode = 1'b0 (default); 1'b1 (set when enabling local IP burn-in test mode, e.g. near-end analog loopback) This signal is used to gate all the scan controls from SOC, thus full-chip scan can be run concurrently with IP burn-in mode.
1	0h RW	**BSCAN_EXCLUDE:** BSCAN exclude
0	0h RW	**SIDEBAND_OR_STAP_SELECT:** This bit selects IOSF SB or STAP access 1 - STAP 0 - IOSF SB. Default selection IOSF SB.

31.7 REF_CTL_REG Register (REF_CTL_REG)—Offset 18h

Reference Control Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
31:9	0h RW	**RESERVED1:** Reserved
8	0h RW	**TX_8XCLKBYPASS:** Bypass 8x clock path and send the data for 4x path
7:6	0h RW	**RESERVED0:** Reserved
5:1	0h RW	**VREG_REFSEL:** RegulatorReference select between 1/4 to 3/4 of reference voltage
0	1h RW	**VREG_REFEN:** Regulator Reference Enable

31.8 DLY_ADJ_REG Register (DLY_ADJ_REG)—Offset 1Ch

Delay Adjustment Register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW	**RESERVED0:** Reserved
19:16	0h RW	**ADJ_DLY_4XQ_DFE:** Delay adjustment for 4x Quad-Phase clock going to DFE
15:12	0h RW	**ADJ_DLY_4XI_DFE:** Delay adjustment for 4x In-Phase clock going to DFE
11:8	0h RW	**ADJ_DLY_8X:** Delay adjustment for 8x lock
7:4	0h RW	**ADJ_DLY_4XQ:** Delay adjustment for 4x qual -phase clock
3:0	0h RW	**ADJ_DLY_4XI:** Delay adjustment for 4x in -phase clock

31.9 HS_RCOMP_READOUT Register (HS_RCOMP_READOUT)—Offset 20h

HS RCOMP Readout Register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO/V	**RESERVED2:** Reserved
15	0h RO/V	**HS_CALIB_EXIT_TYPE:** HS_CALIB_EXIT_TYPE
14:13	0h RO/V	**RESERVED1:** Reserved
12:8	0h RO/V	**FINAL_HS_RCOMP_CODE_OUT:** Final HS RCOMP updated value: final_hs_rcomp_code_out[4:0]. This final code going into the AFE.
7	0h RO/V	**HS_CALIB_EXIT_STATUS:** CALIB_EXIT_STATUS: 0=)Normal exit; 1=)Error condition

Bit Range	Default & Access	Field Name (ID): Description
6	0h RO/V	**HS_RCOMP_DONE_STICKY:** HS RCOMP Calibration and update complete
5	0h RO/V	**RESERVED0:** Reserved
4:0	0h RO/V	**FINAL_HS_RCOMP_CODE_INT:** Final HS RCOMP updated value: final_hs_rcomp_code_out[4:0]. This field is expected to hold calibrated value before update. Need to fix the RTL.

31.10 LP_P_RCOMP_READOUT Register (LP_P_RCOMP_READOUT)—Offset 24h

LP P RCOMP Readout Register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO/V	**RESERVED2:** Reserved
24:21	0h RO/V	**FINAL_LP_P_RCOMP_SLEW:** Final LP P RCOMP slew rate updated value: final_lp_p_rcomp_slew[3:0]
20:12	0h RO/V	**RESERVED1:** Reserved
11:8	0h RO/V	**FINAL_LP_P_RCOMP_CODE_OUT:** Final LP P RCOMP updated value: final_lp_p_rcomp_code_out[3:0]. This final code going into the AFE.
7	0h RO/V	**LP_P_RCOMP_DONE_STICKY:** LP_P_RCOMP_DONE_STICKY
6	0h RO/V	**LP_P_CALIB_EXIT_TYPE:** LP_P_CALIB_EXIT_TYPE
5	0h RO/V	**LP_P_CALIB_EXIT_STATUS:** CALIB_EXIT_STATUS: 0=)Normal exit; 1=)Error condition
4	0h RO/V	**RESERVED0:** Reserved
3:0	0h RO/V	**FINAL_LP_P_RCOMP_CODE_INT:** Final LP P RCOMP calibrated value. This field is expected to hold calibrated value before update. Need to fix the RTL.

31.11 LP_N_RCOMP_READOUT Register (LP_N_RCOMP_READOUT)—Offset 28h

LP N RCOMP Readout Register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:25	0h RO/V	**RESERVED2:** Reserved
24:21	0h RO/V	**FINAL_LP_N_RCOMP_SLEW:** Final LP N RCOMP slew rate updated value: final_lp_n_rcomp_slew[3:0]
20:12	0h RO/V	**RESERVED1:** Reserved
11:8	0h RO/V	**FINAL_LP_N_RCOMP_CODE_OUT:** Final LP N RCOMP updated value: final_lp_n_rcomp_code_out[3:0]. This final code going into the AFE.
7	0h RO/V	**LP_N_RCOMP_DONE_STICKY:** LP_N_RCOMP_DONE_STICKY
6	0h RO/V	**LP_N_CALIB_EXIT_TYPE:** LP_N_CALIB_EXIT_TYPE
5	0h RO/V	**LP_N_CALIB_EXIT_STATUS:** CALIB_EXIT_STATUS: 0=)Normal exit; 1=)Error condition
4	0h RO/V	**RESERVED0:** Reserved
3:0	0h RO/V	**FINAL_LP_N_RCOMP_CODE_INT:** Final LP N RCOMP calibrated value. This field is expected to hold calibrated value before update. Need to fix the RTL.

31.12 RCOMP_LP_SLEW_CFG Register (RCOMP_LP_SLEW_CFG)—Offset 2Ch

RCOMP LP Slew Configuration Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:14	0h RW	**RESERVED1:** Reserved
13:10	0h RW	**LP_DRV_STRG_SEL:** LP Driver Strength Select btw 150, 200, 300, 600 ohm
9:6	0h RW	**LP_P_RCOMP_SLEW_OVR_PUP:** LP RCOMP Override code for Pull-UP slew
5	0h RW	**RESERVED0:** Reserved
4:1	0h RW	**LP_N_RCOMP_SLEW_OVR_PDN:** LP RCOMP Override code for Pull-down slew

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**LP_RCOMP_SLEW_OVR_ENABLE:** LP RCOMP slew Override valid

31.13 CLK_MONITOR_PORT_CONTROL Register (CLK_MONITOR_PORT_CONTROL)—Offset 30h

CLOCK MONITOR PORT CONTROL

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RW	**RESERVED0:** RESERVED
2	0h RW	**CLK_MONITOR_PORT_EN:** Enable the Clock Monitor debug port. 0 = Disable Clock Monitor port. Tie it to '0', so that there is no toggle 1 = Enable the Clock Monitor debug port and route the appropriate clock
1:0	0h RW	**CLK_MONITOR_PORT_CLK_SEL:** Clock Monitor Clock Select to route the required clock to Debug port 00 : DDRCLK = CLK_8x select 01 : DDR_CLK_I = CLK4x_i Select 10 : DDR_CLK_Q = CLK4x_q Select 11: Byte Clock select

31.14 HS_RCOMP_CFG Register (HS_RCOMP_CFG)—Offset 34h

HS RCOMP Configuration Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RW	**RESERVED5:** Reserved
22	0h RW	**RESERVED4:** Was used for HS_NO_WAIT_UPDATE : Forced update mode for HS ; Does not wait for any device state;
21	0h RW	**RESERVED3:** Was used for HS_RCOMP_UPDATE: 0b = RCOMP is updated if the clock lane is in LP11 or ULPS state; 1b = RCOMP is updated if all the data lanes are in LP11 or ULPS state.
20:19	0h RW	**HS_TOGGLE_LIMIT_CREG_ENC:** HS_TOGGLE_LIMIT_CREG_ENC: HS state machine toggle limit. Number of times RCOMP state machine will toggle before exiting. 2'b00: 6 toggles; 2'b01: 4 toggles; 2'b10: 8 toggles; 2'b11: 10 toggles
18:16	0h RW	**RESERVED2:** Reserved

Bit Range	Default & Access	Field Name (ID): Description
15:9	0h RW	**RESERVED1:** Reserved
8	0h RW	**HS_RCOMP_OVERRIDE_VALID:** HS override enable
7:6	0h RW	**RESERVED0:** Reserved
5:1	0h RW	**HS_RCOMP_OVERRIDE_CODE:** HS override values
0	1h RW	**HS_RCOMP_ENABLE:** Enable HS calibration

31.15 LP_RCOMP_CFG Register (LP_RCOMP_CFG)—Offset 38h

LP RCOMP Configuration Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
31:23	0h RW	**RESERVED5:** Reserved
22	0h RW	**RESERVED4:** Was used for LP_NO_WAIT_UPDATE : Forced update mode for LP ; Does not wait for any device state;
21	0h RW	**RESERVED3:** Was used for LP_RCOMP_UPDATE: 0b = RCOMP is updated if the clock lane is in LP11 or ULPS state; 1b = RCOMP is updated if all the data lanes are in LP11 or ULPS state.
20:19	0h RW	**LP_TOGGLE_LIMIT_CREG_ENC:** LP_TOGGLE_LIMIT_CREG_ENC: HS state machine toggle limit. Number of times RCOMP state machine will toggle before exiting. 2'b00: 6 toggles; 2'b01: 4 toggles; 2'b10: 8 toggles; 2'b11: 10 toggles
18:16	0h RW	**RESERVED2:** Reserved
15:11	0h RW	**RESERVED1:** Reserved
10:7	0h RW	**LP_P_RCOMP_OVERRIDE_CODE:** LP RCOMP Overide code for Pull-UP strength
6	0h RW	**RESERVED0:** Reserved
5:2	0h RW	**LP_N_RCOMP_OVERRIDE_CODE:** LP RCOMP Overide code for Pull-down strength
1	0h RW	**LP_RCOMP_OVERRIDE_VALID:** LP override enable
0	1h RW	**LP_RCOMP_ENABLE:** Enable LP calibration

31.16 RCOMP_STAT Register (RCOMP_STAT)—Offset 3Ch

RCOMP Status register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO/V	**RESERVED3:** Reserved
15	0h RO/V	**LP_TXP_RCOMP_COUNTUP:** afe_dfx_lptxpcountup_ff
14	0h RO/V	**LP_TXN_RCOMP_COUNTUP:** afe_dfx_lptxncountup_ff
13	0h RO/V	**HS_RCOMP_COUNTUP:** afe_dfx_hsrcompcountup_ff
12:9	0h RO/V	**RESERVED2:** Was used for LP_P_RCOMP_CODE_MUX : lp_n_rcomp_code_out_mux. Make this reserved.
8:5	0h RO/V	**RESERVED1:** Was used for LP_N_RCOMP_CODE_MUX : lp_p_rcomp_code_out_mux. Make this reserved.
4:0	0h RO/V	**RESERVED0:** Was used for HS_RCOMP_CODE_OUT_MUX : hs_rcomp_code_out_mux. Make this reserved.

31.17 RCCRCOMP Register (RCCRCOMP)—Offset 40h

Rcomp Leg Control

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:4	0h RW	**RESERVED_3:** Reserved
3	0h RW	**RESERVED2:** Was used for HS_RCOMP_CLK_EN : mdf_dphy_rcompclken. Now Reserved
2	0h RW	**RESERVED1:** Was used for HS_RCOMP_FLIS_OVRD : mdf_dphy_rcompflisovrd
1:0	0h RW	**RESERVED0:** RESERVED

31.18 RCOMPCTL Register (RCOMPCTL)—Offset 44h

RCOMP Control

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**RESERVED_3:** Reserved
30	0h RW	**TXDPHY_TESTHSTXSTATICLEG_DISABLE:** DFX signal to disable HS static leg
29	0h RW	**TXDPHY_TESTLPPDSTATICLEG_DISABLE:** DFX signal to disable PD static leg
28	0h RW	**TXDPHY_TESTLPPUSTATICLEG_DISABLE:** DFX signal to disable PU static leg
27	0h RW	**LP_RCOMP_CONTROL_SELECT:** This signal is used to select between functional LP rcomp control and dfx LP rcomp control (enable RCOMP lane manual RCOMP cal test)
26	0h RW	**HS_RCOMP_CONTROL_SELECT:** This signal is used to select between functional HS rcomp control and dfx HS rcomp control
25	0h RW	**RESEREVED:** Reserved
24:21	0h RW	**LPTXNSTRCAL:** DFX override for LP N-strength calibration bits
20	0h RW	**RESERVED0:** Reserved
19:16	0h RW	**LPTXPSTRCAL:** DFX override for LP P-strength calibration bits (override lp_p_next_code to RCOMP lane)
15	0h RW	**LPTXNCOMPEN:** DFX override value for LP N Analog rcomp enable
14	0h RW	**LPTXPCOMPEN:** DFX override value for LP P Analog rcomp enable
13:11	0h RW	**RCOMPTARGET:** DFX override for RCOMP Target level
10:6	0h RW	**HSRCOMPCAL:** DFX override for HS rcomp calibration bits
5	0h RW	**HSRCOMPEN:** DFX override for HS rcomp enable
4	0h RW	**RCOMPSTART:** Customer Trigger/DFX HS start signal to trigger RCOMP FSM. By default this bit is auto cleared when RCOMPSTART_AUTO_CLR_DIS = 0 else auto clear is disabled
3	0h RW	**RCOMPSTART_AUTO_CLR_DIS:** This bit is used to disable the RCOMPSTART auto clear feature. By default RCOMPSTART_AUTO_CLR_DIS = 0 : auto-clear is enabled RCOMPSTART_AUTO_CLR_DIS = 1 : auto clear disabled NOTE : This is valid only via APB registers. The RCOMPSTART Auto-Clear feature is not available in the mirror STAP register

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**LP_RCOMP_OVERRIDE_SELECT:** This bit is used to select between functional LP Rcomp code and DFX LP rcomp code
1	0h RW	**HSREGENRCOMP:** DFX override to enable HS regulator for Rcomp lane
0	0h RW	**HS_RCOMP_OVERRIDE_SELECT:** This signal is used to select between functional HS rcomp control and dfx rcomp controls

31.19 PWRMNG_OVR_CTL Register (PWRMNG_OVR_CTL)—Offset 48h

Power Management Override Control Register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RW	**RESERVED0:** Reserved
29	0h RW	**PHY_RST_B_OVR:** PHY RESET RESET BAR VALUE
28	0h RW	**RCOMP_RESET_OVR:** RCOMP RESET VALUE: (ACTIVE HIGH) This field is only used if 0: RCOMP is not IN RESET 1: RCOMP is FORCED to RESET
27	0h RW	**BITRATECLK_ACK_OVR:** BITRATE Clock ACK OVERRIDE VALUE
26	0h RW	**INDUCE_LP00_OVR:** Induce LP00 VALUE
25	0h RW	**PHYREADY_B_OVR:** PHY Ready Override VALUE
24	0h RW	**PWRSTABLE_OVR:** Power Stable Override VALUE
23	0h RW	**AFE_FWEN_B_OVR:** AFE Frirewall Enable OVERRIDE VALUE
22	0h RW	**LPLATCHENB_OVR:** LPLATCHENB OVERRIDE VALUE
21	0h RW	**LP00_LPWRMODE_OVR:** LP00_LPWRMODE OVERRIDE VALUE (UNUSED)
20	0h RW	**LP11_LPWRMODE_OVR:** LP11_LPWRMODE OVERRIDE VALUE (UNUSED)
19	0h RW	**LPWAKE_OVR:** Low power Wake OVERRIDE VALUE
18	0h RW	**MIPI_MODE_OVR:** MIPI MODE OVERRIDE VALUE

Bit Range	Default & Access	Field Name (ID): Description
17	0h RW	**BITRATECLK_REQ_OVR:** BITRATE Clock OVERRIDE REQ VALUE
16	0h RW	**PHYCLK_REQ_OVR:** PHY Clock OVERRIDE REQ VALUE
15	0h RW	**PHY_RST_B_OVR_EN:** PHY RESET OVERRIDE ENABLE: 0: phy_rst_b reset is an input to the DPHY IP 1: phy_rst_b reset is overriden by PHY_RST_B_OVR
14	0h RW	**RCOMP_RESET_OVR_EN:** RCOMP RESET OVERRIDE ENABLE: 0: Rcomp Reset genmerated out of FSM is not overriden 1: Rcomp Reset genmerated out of FSM is overriden by RCOMP_RESET_OVR
13	0h RW	**BITRATECLK_ACK_OVR_EN:** BITRATE Clock ACK OVERRIDE ENABLE: 0: Input bitrateclk_ack is not overriden 1: Input bitrateclk_ack is overriden, value specified by BITRATECLK_ACK_OVR
12	0h RW	**S0I3_ENTRY_DISABLE:** s0i3 Entry disable 0: s0i3 entry will ocurr every time ulps state is detected while in normal run mode 1: s0i3 entry is disabled
11	0h RW	**INDUCE_LP00_OVR_EN:** Iduce LP 00 (Warm reset) Override Enable 0: Induce LP00 from conroller is not overriden 1: induce_lp00 is overriden, value specified by INDUCE_LP00_OVR
10	0h RW	**SW_SEQ_EN_OVR_EN:** Software Sequence Enable Override Enable 0: Power Sequence is handled by Power Hardwared FSM within MIPIO 1: Controller Software Power Sequence is Enable
9	0h RW	**PHYREADY_B_OVR_EN:** PHY Ready Override Enable 0: phyready_b is not overriden 1: phyready_b is overriden, value specified by PHYREADY_B_OVR
8	0h RW	**PWRSTABLE_OVR_EN:** Power Stable Override Enable 0: pwrstable is not overriden 1: pwrstable is overriden, value specified by PWRSTABLE_OVR
7	0h RW	**AFE_FWEN_B_OVR_EN:** AFE Frirewall Enable OVERRIDE ENABLE: 0: afe_fwen_b is not overriden 1: afe_fwen_b is overriden, value specified by AFE_FWEN_B_OVR
6	0h RW	**LPLATCHENB_OVR_EN:** LPLATCHENB OVERRIDE ENABLE: 0: lplatchenb is not overriden 1: lplatchenb is overriden, value specified by LPLATCHENB_OVR
5	0h RW	**LP00_LPWRMODE_OVR_EN:** LP00_LPWRMODE OVERRIDE ENABLE: (UNUSED) 0: lp00_lpwrmode is not overriden 1: lp00_lpwrmode is overriden, value specified by LP00_LPWRMODE_OVR
4	0h RW	**LP11_LPWRMODE_OVR_EN:** LP11_LPWRMODE OVERRIDE ENABLE: (UNUSED) 0: lp11_lpwrmode is not overriden 1: lp11_lpwrmode is overriden, value specified by LP11_LPWRMODE_OVR
3	0h RW	**LPWAKE_OVR_EN:** Low power Wake OVERRIDE ENABLE: 0: lpwake is not overriden 1: lpwake is overriden, value specified by LPWAKE_OVR
2	0h RW	**MIPI_MODE_OVR_EN:** MIPI MODE OVERRIDE ENABLE: 0: mipi_mode is not overriden 1: mipi_mode is overriden, value specified by MIPI_MODE_OVR
1	0h RW	**BITRATECLK_REQ_OVR_EN:** BITRATE Clock REQUEST OVERRIDE ENABLE: 0: bitrateclk_req is not overriden 1: bitrateclk_req is overriden, value specified by BITRATECLK_REQ_OVR
0	0h RW	**PHYCLK_REQ_OVR_EN:** PHY Clock REQUEST OVERRIDE ENABLE: 0: phyclk_req is not overriden 1: phyclk_req is overriden, value specified by PHYCLK_REQ_OVR

31.20 PWRMNG_STATUS Register (PWRMNG_STATUS)—Offset 4Ch

Power Management STATUS Register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RO	**RESERVED0:** Reserved
6	0h RO	**BITRATECLK_CLKACK:** CLOCK ACK for BITRATE CLOCK
5	0h RO	**PHYCLK_CLKACK:** CLOCK ACK for PHY CLOCK
4	0h RO	**PORT1_LATCHOUTCP_STAT:** Port1 LATCHOUTCP STATUS: Reflects current Value of latcnoutcp from Port0 AFE From Controller
3	0h RO	**PORT0_LATCHOUTCP_STAT:** Port0 LATCHOUTCP STATUS: Reflects current Value of latcnoutcp from Port0 AFE From Controller
2	0h RO	**LATCHENB_STAT:** LATCHENB STATUS: Reflects current Value of LATCHENB After Override MUX
1	0h RO	**AFE_FWEN_B_STAT:** AFE Firewall Enable STATUS: Reflects current Value of AFE_FWEN_B After Override MUX
0	0h RO	**MIPI_MODE_STAT:** MIPI_MODE STATUS: Reflects current Value of MIPI_MODE After Override MUX

31.21 GLOBAL_GLK_RESERVED3 Register (GLOBAL_GLK_RESERVED3)—Offset 50h

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**RESERVED0**

31.22 GLOBAL_RESERVED1 Register (GLOBAL_RESERVED1)—Offset 54h

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.23 TX_DFX_TOP_CFG_REG Register (TX_DFX_TOP_CFG_REG)—Offset 58h

TX DFX Top Config Register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:6	0h RW	**RESERVED0:** Reserved
5	0h RW	**DFX_DPHY_LATCH_EN:** DFX_DPHY_LATCH EN. Enable latch in AFE to allow data to pass through
4	0h RW	**TXDPHY_TESTLPCDDISABLE_N:** Disable contention detection, Goes to AFE. Used to disable LPCD in debug mode
3	0h RW	**BSCAN_LATCH_RESETB:** In bscan mode, this signal will override the itx_latchresetb
2	0h RW	**BSCAN_LATCH_SELECT:** In bscan mode, use this signal to enable bscan_latchresetb override
1	0h RW	**ALL_1:** All 1. Only required for Tx Path
0	0h RW	**ALL_0:** All 0. Only required for Tx Path

31.24 TX_DFX_TOP_RESERVED1 Register (TX_DFX_TOP_RESERVED1)—Offset 5Ch

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.25 TX_DFX_TOP_RESERVED2 Register (TX_DFX_TOP_RESERVED2)—Offset 60h

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.26 TX_DFX_TOP_RESERVED3 Register (TX_DFX_TOP_RESERVED3)—Offset 64h

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.27 TX_DFX_TOP_RESERVED4 Register (TX_DFX_TOP_RESERVED4)—Offset 68h

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.28 TX_DFX_TOP_RESERVED5 Register (TX_DFX_TOP_RESERVED5)—Offset 6Ch

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.29 TX_DFX_TOP_RESERVED6 Register (TX_DFX_TOP_RESERVED6)—Offset 70h

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.30 TX_DFX_TOP_RESERVED7 Register (TX_DFX_TOP_RESERVED7)—Offset 74h

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.31 TX_DFX_TOP_RESERVED8 Register (TX_DFX_TOP_RESERVED8)—Offset 78h

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.32 TX_DFX_TOP_RESERVED9 Register (TX_DFX_TOP_RESERVED9)—Offset 7Ch

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.33 TX_DFX_PORT0_CTL_REG Register (TX_DFX_PORT0_CTL_REG)—Offset 80h

TX DFX Port-0 Level Control Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RW	**RESERVED1:** Reserved
17	0h RW	**CONTENTION_DETECTION_DATAN:** CD DATA DN: Drive Contention data on LP DN pin from FLIS DFX

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW	**CONTENTION_DETECTION_DATAP:** CD DATA DP: Drive Contention data on LP DP pin from FLIS DFX
15	0h RW	**CONTENTION_DETECTION_MODE:** CD MODE. Enable Contention Detect Mode in testing
14:11	0h RW	**RESERVED0:** Reserved
10:9	0h RW	**HS_CLK_SEL:** High speed clock select for DFX mode
8:7	0h RW	**HS_DATA_SEL:** High speed data select for DFX mode
6:5	0h RW	**LP_CLK_SEL:** Low power clock select for DFX mode
4:2	0h RW	**LP_DATA_SEL:** Low power data select for DFX mode
1	0h RW	**LP_IO_CONTROL_SELECT:** DFX override for LP TX DATA
0	0h RW	**HS_IO_CONTROL_SELECT:** DFX override for HS TX DATA

31.34 TX_DFX_PORT0_CD_COUNTUP_STATUS Register (TX_DFX_PORT0_CD_COUNTUP_STATUS)—Offset 84h

CD Countup Status

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**RESERVED0:** RESERVED
11	0h RO	**LP_CD_D1_0_META_FLOP_CORE:** LPCD High Test Mode Meta flop output This is the LPCD High signal sync'ed with Tx Escape Clock
10	0h RO	**LP_CD_D0_0_META_FLOP_CORE:** LPCD Low Test Mode Meta flop output This is the LPCD Low signal sync'ed with Tx Escape Clock
9:6	0h RO	**ALPCDHIGH_TEST:** LPCD High Test Mode output
5:2	0h RO	**ALPCDLOW_TEST:** LPCD Low Test Mode output
1	0h RO	**ALPCDHIGH_0:** LPCD contention high
0	0h RO	**ALPCDLOW_0:** LPCD contention low

31.35 TX_DFX_PORT0_RCC_CFG_REG Register (TX_DFX_PORT0_RCC_CFG_REG)—Offset 88h

RCC Config Register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RW	**RESERVED3:** Reserved
17	0h RW	**HS_RCC_TX_START:** HS_RCC_TX_Start
16	0h RW	**HS_RCC_TX_ENABLE:** HS_RCC_TX_Enable
15:12	0h RW	**TEST_DATA_PATTERN:** Test_Data_Pattern
11	0h RW	**RESERVED2:** Reserved
10	0h RW	**RESERVED1:** Reserved
9	0h RW	**RESERVED0:** Reserved
8	0h RW	**FINITE_TEST:** Finite_Test
7:0	0h RW	**MAX_COUNT:** 8-bit max count for test IO TEST ENGINE RCC

31.36 TX_DFX_PORT0_ACIO_CFG_REG Register (TX_DFX_PORT0_ACIO_CFG_REG)—Offset 8Ch

ACIO Loopback Config Register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**DATA_MASK:** Data Mask IO TEST ENGINE
27:12	0h RW	**TEST_DATA_PATTERN:** Test Data Pattern IO TEST ENGINE

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**RESERVED2:** Reserved
10	0h RW	**RESERVED1:** Reserved
9	0h RW	**RESERVED0:** Reserved
8	0h RW	**FINITE_TEST:** Finite_Test
7:0	0h RW	**MAX_COUNT:** 8-bit max count for test IO TEST ENGINE

31.37 TX_DFX_PORT0_ACIOSS_REG Register (TX_DFX_PORT0_ACIOSS_REG)—Offset 90h

ACIO Loopback Start and Status Register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW	**RESERVED2:** Reserved
19	0h RW	**LB_FIFO_RST_B:** Resets the loopback FIFO read and write pointers.
18	0h RW	**LP_CDFELB_SEL:** select between low and high contention
17	0h RW	**LP_FELB_SEL:** select between contention and rxpath
16	0h RW	**LP_RXFELB_SEL:** Rx Far End loopback select. At portlevel selects between toggle detector and rx path for lane0
15	0h RW	**RESERVED1:** Reserved
14	0h RW	**PRG_DEL:** prg_del
13	0h RW	**CAL_IOTE_DELA:** cal_iote_dela
12	0h RW	**SYNCFLOPBYPASSMODEEN_N:** Enable to bypass DSI sync flop stage
11	0h RW	**SYNCFLOPBYPASSMODE_CTRL:** Control to allow bypass sync flop enable to pass to AFE and override DSI controller bypass flop muxing enable bit=dpa_mio_portctrl_a[8]
10	0h RW	**SERIALIZER_ENABLE:** This will enable rd_en on TXPORT sync FIFO and will also reset TX sync FIFO

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**RESERVED0:** TXPORT_LOOPBACK, Reserved, MUST be 1'b0 always
8:5	0h RW	**DELAY1:** These bits delay comparison if the received data takes more than the default number of cycles to reach the comparator
4	0h RW	**ACIO_START_A:** Start ACIO Loopback test using IOTE : 0-Stop test, 1-Start test
3	0h RW	**ACIOLB_A_RST:** Reset for cal_iote_del_a
2	0h RW	**LB_FIFO_PTR_SEP:** Loopback FIFO pointer separation between write and read pointers. 0 - read pointer delayed 1 cycle, 1 - read pointer delayed 2 cycles
1	0h RW	**ACIO_MODE_CLN_HSREQ_OVR:** ACIO Mode Clock Lane High Speed Request Override Control. This bit must be set first before setting the Start bits When 1'b0: Functional MODE is selected for CLN_HSREQ When 1'b1: CLN_HSREQ is forced to 1'b1 for ACIO LB operation
0	0h RW	**ACIOLBEN:** AC IO loopback enable --> Connect to Internal ACIO Loopback Enable (sent to AFE to enable loopback circuitry)

31.38 TX_DFX_PORT0_ACIOERR_REG Register (TX_DFX_PORT0_ACIOERR_REG)—Offset 94h

ACIO Loopback Error Register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**RESERVED_2:** Reserved
28	0h RO	**IOTE_START_FF:** This bit indicates IOTES ENGINE A has started
27:20	0h RO	**TRANSMIT_DATA:** Generated data from IO Test Engine to check if the IOTE A did indeed start
19:16	0h RO	**DATA_ERROR_RESULT:** Data0_Error_Result from IOTE ENGINE A
15	0h RO	**ACIO_COMPLETE:** This indicates that ACIO loopback test using IOTE Engine A is complete
14:8	0h RO	**RESERVED0:** Reserved
7:0	0h RO	**MAX_COUNT:** 8-bit max count number when the test is done for IOTE ENGINE A

31.39 TX_DFX_PORT0_DELAY_CALIB_REG Register (TX_DFX_PORT0_DELAY_CALIB_REG)—Offset 98h

Delay Calibration Register

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RO	**RESERVED0:** Reserved
4	0h RO	**COMP_STATA:** comp_stata
3:0	0h RO	**CAL_DELAY1:** cal_delay1

31.40 TX_DFX_PORT0_RXLPBK_DATA Register (TX_DFX_PORT0_RXLPBK_DATA)—Offset 9Ch

RX Loopback Data Register

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RX_LPBK_DATA1:** rx_lpbk_data1 (CLK A) --> Use SIPO instead of Receiver data

31.41 TX_DFX_PORT0_CLANE_CTL_REG Register (TX_DFX_PORT0_CLANE_CTL_REG)—Offset A0h

TX DFX Clock Lane Control Reg

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RW	**RESERVED0:** Reserved
6:3	0h RW	**TIME_SKEW_ADJUST_CLK:** Lane Timing Skew adjustability between CLK/DATA for DSI with 4-bit control for each lane: ~25ps resolution
2	0h RW	**HS_TX_REGEN_CLK:** DFX override for HS TX REG enable
1	0h RW	**HS_TX_EN_CLK:** DFX override for HS TX enable
0	0h RW	**LP_TX_EN_CLK:** DFX override for LP TX enable

31.42 TX_DFX_PORT0_RESERVED1 Register (TX_DFX_PORT0_RESERVED1)—Offset A4h

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.43 TX_DFX_PORT0_RESERVED2 Register (TX_DFX_PORT0_RESERVED2)—Offset A8h

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.44 TX_DFX_PORT0_RESERVED3 Register (TX_DFX_PORT0_RESERVED3)—Offset ACh

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.45 TX_DFX_PORT0_DLANE0_CTL_REG Register (TX_DFX_PORT0_DLANE0_CTL_REG)—Offset B0h

TX DFX Data Lane-0 Control Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RW	**RESERVED0:** Reserved
17	0h RW	**LP_RX_EN:** LP RX DATA Lane-0 CNTL TESTMODE. DFX override for LP RX CNTL
16:9	0h RW	**COMP_SEL_OVRD:** (DSI_PIPEA_SIPO0) compare_sel_ovrd: Overriden value for symbol
8	0h RW	**CP_STAT:** (DSI_PIPEA_SIPO0) cp_stat: Enable symbol value from sync_align logic output
7	0h RW	**CP_STAT_OVRD:** (DSI_PIPEA_SIPO0) cp_stat_ovrd: 1 - select overriden value, 0 - select value from sync_align logic
6:3	0h RW	**TIME_SKEW_ADJUST_DATA:** Lane 0-3 Timing Skew adjustability between CLK/ DATA for DSI with 4-bit control for each lane: ~25ps resolution
2	0h RW	**HS_TX_REGEN:** DFX override for HS TX DATA0 REG enable
1	0h RW	**HS_TX_EN:** DFX override for HS TX DATA0 enable
0	0h RW	**LP_TX_EN:** DFX override for LP TX DATA0 enable

31.46 TX_DFX_PORT0_DLANE0_STAT_REG Register (TX_DFX_PORT0_DLANE0_STAT_REG)—Offset B4h

TX DFX Data Lane-0 Status Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**RESERVED0:** Reserved
7:0	0h RO	**INTLPBK_COMP_STAT:** Compare status from SIPO0 - Pipe A

31.47 TX_DFX_PORT0_DLANE1_CTL_REG Register (TX_DFX_PORT0_DLANE1_CTL_REG)—Offset B8h

TX DFX Data Lane-1 Control Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RW	**RESERVED1:** Reserved
17	0h RW	**LP_RX_EN:** LP RX DATA Lane-1 CNTL TESTMODE. DFX override for LP RX CNTL
16:9	0h RW	**COMP_SEL_OVRD:** (DSI_PIPEA_SIPO0) compare_sel_ovrd: Overriden value for symbol
8	0h RW	**CP_STAT:** (DSI_PIPEA_SIPO0) cp_stat: Enable symbol value from sync_align logic output
7	0h RW	**CP_STAT_OVRD:** (DSI_PIPEA_SIPO0) cp_stat_ovrd: 1 - select overriden value, 0 - select value from sync_align logic
6:3	0h RW	**RESERVED0:** RESERVED
2	0h RW	**HS_TX_REGEN:** DFX override for HS TX DATA1 REG enable
1	0h RW	**HS_TX_EN:** DFX override for HS TX DATA1 enable
0	0h RW	**LP_TX_EN:** DFX override for LP TX DATA1 enable

31.48 TX_DFX_PORT0_DLANE1_STAT_REG Register (TX_DFX_PORT0_DLANE1_STAT_REG)—Offset BCh

TX DFX Data Lane-1 Status Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**RESERVED0:** Reserved
7:0	0h RO	**INTLPBK_COMP_STAT:** Compare status from SIPO1 - Pipe A

31.49 TX_DFX_PORT0_DLANE2_CTL_REG Register (TX_DFX_PORT0_DLANE2_CTL_REG)—Offset C0h

TX DFX Data Lane-2 Control Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RW	**RESERVED1:** Reserved
17	0h RW	**LP_RX_EN:** LP RX DATA Lane-2 CNTL TESTMODE. DFX override for LP RX CNTL
16:9	0h RW	**COMP_SEL_OVRD:** (DSI_PIPEA_SIPO0) compare_sel_ovrd: Overriden value for symbol
8	0h RW	**CP_STAT:** (DSI_PIPEA_SIPO0) cp_stat: Enable symbol value from sync_align logic output
7	0h RW	**CP_STAT_OVRD:** (DSI_PIPEA_SIPO0) cp_stat_ovrd: 1 - select overriden value, 0 - select value from sync_align logic
6:3	0h RW	**RESERVED0:** RESERVED
2	0h RW	**HS_TX_REGEN:** DFX override for HS TX DATA2REG enable
1	0h RW	**HS_TX_EN:** DFX override for HS TX DATA2enable
0	0h RW	**LP_TX_EN:** DFX override for LP TX DATA2enable

31.50 TX_DFX_PORT0_DLANE2_STAT_REG Register (TX_DFX_PORT0_DLANE2_STAT_REG)—Offset C4h

TX DFX Data Lane-2 Status Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**RESERVED0:** Reserved
7:0	0h RO	**INTLPBK_COMP_STAT:** Compare status from SIPO2 - Pipe A

31.51 TX_DFX_PORT0_DLANE3_CTL_REG Register (TX_DFX_PORT0_DLANE3_CTL_REG)—Offset C8h

TX DFX Data Lane-3 Control Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RW	**RESERVED1:** Reserved
17	0h RW	**LP_RX_EN:** LP RX DATA Lane-3 CNTL TESTMODE. DFX override for LP RX CNTL
16:9	0h RW	**COMP_SEL_OVRD:** (DSI_PIPEA_SIPO0) compare_sel_ovrd: Overriden value for symbol
8	0h RW	**CP_STAT:** (DSI_PIPEA_SIPO0) cp_stat: Enable symbol value from sync_align logic output
7	0h RW	**CP_STAT_OVRD:** (DSI_PIPEA_SIPO0) cp_stat_ovrd: 1 - select overriden value, 0 - select value from sync_align logic
6:3	0h RW	**RESERVED0:** RESERVED
2	0h RW	**HS_TX_REGEN:** DFX override for HS TX DATA3REG enable
1	0h RW	**HS_TX_EN:** DFX override for HS TX DATA3enable
0	0h RW	**LP_TX_EN:** DFX override for LP TX DATA3enable

31.52 TX_DFX_PORT0_DLANE3_STAT_REG Register (TX_DFX_PORT0_DLANE3_STAT_REG)—Offset CCh

TX DFX Data Lane-3 Status Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**RESERVED0:** Reserved
7:0	0h RO	**INTLPBK_COMP_STAT:** Compare status from SIPO3 - Pipe A

31.53 TX_DFX_PORT0_LANE_RESERVED5 Register (TX_DFX_PORT0_LANE_RESERVED5)—Offset D0h

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.54 TX_DFX_PORT0_LANE_RESERVED6 Register (TX_DFX_PORT0_LANE_RESERVED6)—Offset D4h

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.55 TX_DFX_PORT0_LANE_RESERVED7 Register (TX_DFX_PORT0_LANE_RESERVED7)—Offset D8h

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.56 TX_DFX_PORT0_LANE_RESERVED8 Register (TX_DFX_PORT0_LANE_RESERVED8)—Offset DCh

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.57 TX_DFE_PORT0_DLN_CNT_LPX Register (TX_DFE_PORT0_DLN_CNT_LPX)—Offset E0h

lpx count register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: Ah

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	Ah RW	**DLN_CNT_LPX:** period for which a LP state should be driven

31.58 TX_DFE_PORT0_DLN_CNT_HS_PREP Register (TX_DFE_PORT0_DLN_CNT_HS_PREP)—Offset E4h

hs prepare count register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 9h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	9h RW	**DLN_CNT_HS_PREP:** Period for which HS Prepare time should be accomodated

31.59 TX_DFE_PORT0_DLN_CNT_HS_ZERO Register (TX_DFE_PORT0_DLN_CNT_HS_ZERO)—Offset E8h

hs zero count register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 15h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	15h RW	**DLN_CNT_HS_ZERO:** Period for which HS Zero should be driven before sync

31.60 TX_DFE_PORT0_DLN_CNT_HS_TRAIL Register (TX_DFE_PORT0_DLN_CNT_HS_TRAIL)—Offset ECh

hs trail count register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: Bh

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	Bh RW	**DLN_CNT_HS_TRAIL:** Period for which HS trailing should be driven

31.61 TX_DFE_PORT0_DLN_CNT_HS_EXIT Register (TX_DFE_PORT0_DLN_CNT_HS_EXIT)—Offset F0h

hs exit count register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 12h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	12h RW	**DLN_CNT_HS_EXIT:** Period for which HS exit state should be maintained

31.62 TX_DFE_PORT0_DLN_CNT_RX Register (TX_DFE_PORT0_DLN_CNT_RX)—Offset F4h

rx count register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	1h RW	**DLN_CNT_RX:** Counter that controls the assertion of enable on the DPHY

31.63 TX_DFE_PORT0_DLN_SYNC_CNT Register (TX_DFE_PORT0_DLN_SYNC_CNT)—Offset F8h

sync count register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	1h RW	**DLN_SYNC_COUNT:** A timeout value used for Sync error detector logic

31.64 TX_DFE_PORT0_CLN_CNT_HS_TRAIL Register (TX_DFE_PORT0_CLN_CNT_HS_TRAIL)—Offset FCh

clane trail count

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: Bh

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	Bh RW	**CLN_HS_TRAIL:** Wait time in byte clock for the trailing bits

31.65 TX_DFE_PORT0_CLN_CNT_HS_EXIT Register (TX_DFE_PORT0_CLN_CNT_HS_EXIT)—Offset 100h

clane exit count

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 12h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	12h RW	**CLN_HS_EXIT:** Wait time in byte clock for the exit state

31.66 TX_DFE_PORT0_CLN_CNT_LPX Register (TX_DFE_PORT0_CLN_CNT_LPX)—Offset 104h

clane lpx count

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 9h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	9h RW	**CLN_CNT_LPX:** Wait time in byte clock for the LPX

31.67 TX_DFE_PORT0_CLN_CNT_PREP Register (TX_DFE_PORT0_CLN_CNT_PREP)—Offset 108h

clane prepare count

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: Eh

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	Eh RW	**CLN_CNT_PREP:** Wait time in byte clock for the prepare time

31.68 TX_DFE_PORT0_CLN_CNT_ZERO Register (TX_DFE_PORT0_CLN_CNT_ZERO)—Offset 10Ch

clane zero count

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 32h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	32h RW	**CLN_CNT_ZERO:** Wait time in byte clock for the zero state

31.69 TX_DFE_PORT0_CLN_CNT_PLL Register (TX_DFE_PORT0_CLN_CNT_PLL)—Offset 110h

clane PLL count

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**RESERVED0:** Reserved
15:0	0h RW	**CLN_CNT_PLL:** Count value used for the PLL lock time

31.70 TX_DFE_PORT0_CLN_CNT_POST Register (TX_DFE_PORT0_CLN_CNT_POST)—Offset 114h

clane cnt post

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 12h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	12h RW	**CLN_CNT_POST:** Wait time in byte clock for the zero state

31.71 TX_DFE_PORT0_CLN_DLN_ENABLE_OVR Register (TX_DFE_PORT0_CLN_DLN_ENABLE_OVR)—Offset 118h

Clock Lane and Data Lane Enable Override

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** RESERVED
7	0h RW	**AFE_LPRX_EN_CNTRL_SEL:** AFE Low Power Receiver Enable Control Select on the functional Path (i.e before the DFX override) 1 : LP Receiver Enable for Lane 0 is always Enable 0 : Use Data Lane Enables to qualify LP Receiver Enable Control
6	0h RW	**CLN_ENABLE_OVR_SEL:** TX-DFE Clock Lane Enable Override Select Control 1 : Override the Clock Lane Enable from this reg (CLN_ENABL_OVR) 0 : Select the Clock Lane Enable from ORing Data Lane Enable Controls After Override Mux
5:2	0h RW	**DLN_ENABLE_OVR:** TX-DFE Data Lane Enable Control Override Signal DLN_ENABLE_OVR[3:0] DLN_ENABLE_OVR[3] : 1 = Enable the Data Lane-3 ; 0 = Disable Data Lane-3 DLN_ENABLE_OVR[2] : 1 = Enable the Data Lane-2 ; 0 = Disable Data Lane-2 DLN_ENABLE_OVR[1] : 1 = Enable the Data Lane-1 ; 0 = Disable Data Lane-1 DLN_ENABLE_OVR[0] : 1 = Enable the Data Lane-0 ; 0 = Disable Data Lane-0
1	0h RW	**CLN_ENABLE_OVR:** TX-DFE Clock Lane Enable Control Override Signal 1 : Enable the Clock Lane 0 : Disable the Clock Lane [Reset Default]
0	0h RW	**DLN_ENABLE_OVR_SEL:** TX-DFE Data Lane Enable Override Select Control 1 : Override the Data Lane Enable from this reg 0 : Select the Data Lane Enable from PPI or normal TX-DFE control [reset Default]

31.72 TX_DFE_PORT0_RESERVED1 Register (TX_DFE_PORT0_RESERVED1)—Offset 11Ch

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.73 TX_DFX_PORT1_CTL_REG Register (TX_DFX_PORT1_CTL_REG)—Offset 120h

TX DFX Port-1 Level Control Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RW	**RESERVED1:** Reserved
17	0h RW	**CONTENTION_DETECTION_DATAN:** CD DATA DN: Drive Contention data on LP DN pin from FLIS DFX
16	0h RW	**CONTENTION_DETECTION_DATAP:** CD DATA DP: Drive Contention data on LP DP pin from FLIS DFX
15	0h RW	**CONTENTION_DETECTION_MODE:** CD MODE. Enable Contention Detect Mode in testing
14:11	0h RW	**RESERVED0:** Reserved
10:9	0h RW	**HS_CLK_SEL:** High speed clock select for DFX mode
8:7	0h RW	**HS_DATA_SEL:** High speed data select for DFX mode
6:5	0h RW	**LP_CLK_SEL:** Low power clock select for DFX mode
4:2	0h RW	**LP_DATA_SEL:** Low power data select for DFX mode
1	0h RW	**LP_IO_CONTROL_SELECT:** DFX override for LP TX DATA
0	0h RW	**HS_IO_CONTROL_SELECT:** DFX override for HS TX DATA

31.74 TX_DFX_PORT1_CD_COUNTUP_STATUS Register (TX_DFX_PORT1_CD_COUNTUP_STATUS)—Offset 124h

CD Countup Status

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO	**RESERVED0:** RESERVED
11	0h RO	**LP_CD_D1_0_META_FLOP_CORE:** LPCD High Test Mode Meta flop output This is the LPCD High signal sync'ed with Tx Escape Clock
10	0h RO	**LP_CD_D0_0_META_FLOP_CORE:** LPCD Low Test Mode Meta flop output This is the LPCD Low signal sync'ed with Tx Escape Clock
9:6	0h RO	**ALPCDHIGH_TEST:** LPCD High Test Mode output
5:2	0h RO	**ALPCDLOW_TEST:** LPCD Low Test Mode output

Bit Range	Default & Access	Field Name (ID): Description
1	0h RO	**ALPCDHIGH_0:** LPCD contention high
0	0h RO	**ALPCDLOW_0:** LPCD contention low

31.75 TX_DFX_PORT1_RCC_CFG_REG Register (TX_DFX_PORT1_RCC_CFG_REG)—Offset 128h

RCC Config Register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RW	**RESERVED3:** Reserved
17	0h RW	**HS_RCC_TX_START:** HS_RCC_TX_Start
16	0h RW	**HS_RCC_TX_ENABLE:** HS_RCC_TX_Enable
15:12	0h RW	**TEST_DATA_PATTERN:** Test_Data_Pattern
11	0h RW	**RESERVED2:** Reserved
10	0h RW	**RESERVED1:** Reserved
9	0h RW	**RESERVED0:** Reserved
8	0h RW	**FINITE_TEST:** Finite_Test
7:0	0h RW	**MAX_COUNT:** 8-bit max count for test IO TEST ENGINE RCC

31.76 TX_DFX_PORT1_ACIO_CFG_REG Register (TX_DFX_PORT1_ACIO_CFG_REG)—Offset 12Ch

ACIO Loopback Config Register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RW	**DATA_MASK:** Data Mask IO TEST ENGINE
27:12	0h RW	**TEST_DATA_PATTERN:** Test Data Pattern IO TEST ENGINE
11	0h RW	**RESERVED2:** Reserved
10	0h RW	**RESERVED1:** Reserved
9	0h RW	**RESERVED0:** Reserved
8	0h RW	**FINITE_TEST:** Finite_Test
7:0	0h RW	**MAX_COUNT:** 8-bit max count for test IO TEST ENGINE

31.77 TX_DFX_PORT1_ACIOSS_REG Register (TX_DFX_PORT1_ACIOSS_REG)—Offset 130h

ACIO Loopback Start and Status Register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW	**RESERVED2:** Reserved
19	0h RW	**LB_FIFO_RST_B:** Resets the loopback FIFO read and write pointers.
18	0h RW	**LP_CDFELB_SEL:** select between low and high contention
17	0h RW	**LP_FELB_SEL:** select between contention and rxpath
16	0h RW	**LP_RXFELB_SEL:** Rx Far End loopback select. At portlevel selects between toggle detector and rx path for lane0
15	0h RW	**RESERVED1:** Reserved
14	0h RW	**PRG_DEL:** prg_del
13	0h RW	**CAL_IOTE_DELA:** cal_iote_dela
12	0h RW	**SYNCFLOPBYPASSMODEEN_N:** Enable to bypass DSI sync flop stage

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**SYNCFLOPBYPASSMODE_CTRL:** Control to allow bypass sync flop enable to pass to AFE and override DSI controller bypass flop muxing enable bit=dpa_mio_portctrl_a[8]
10	0h RW	**SERIALIZER_ENABLE:** This will enable rd_en on TXPORT sync FIFO and will also reset TX sync FIFO
9	0h RW	**RESERVED0:** TXPORT_LOOPBACK, Reserved, MUST be 1'b0 always
8:5	0h RW	**DELAY1:** These bits delay comparison if the received data takes more than the default number of cycles to reach the comparator
4	0h RW	**ACIO_START_A:** Start ACIO Loopback test using IOTE : 0-Stop test, 1-Start test
3	0h RW	**ACIOLB_A_RST:** Reset for cal_iote_del_a
2	0h RW	**LB_FIFO_PTR_SEP:** Loopback FIFO pointer separation between write and read pointers. 0 - read pointer delayed 1 cycle, 1 - read pointer delayed 2 cycles
1	0h RW	**ACIO_MODE_CLN_HSREQ_OVR:** ACIO Mode Clock Lane High Speed Request Override Control. This bit must be set first before setting the Start bits When 1'b0: Functional MODE is selected for CLN_HSREQ When 1'b1: CLN_HSREQ is forced to 1'b1 for ACIO LB operation
0	0h RW	**ACIOLBEN:** AC IO loopback enable --> Connect to Internal ACIO Loopback Enable (sent to AFE to enable loopback circuitry)

31.78 TX_DFX_PORT1_ACIOERR_REG Register (TX_DFX_PORT1_ACIOERR_REG)—Offset 134h

ACIO Loopback Error Register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RO	**RESERVED_2:** Reserved
28	0h RO	**IOTE_START_FF:** This bit indicates IOTES ENGINE A has started
27:20	0h RO	**TRANSMIT_DATA:** Generated data from IO Test Engine to check if the IOTE A did indeed start
19:16	0h RO	**DATA_ERROR_RESULT:** Data0_Error_Result from IOTE ENGINE A
15	0h RO	**ACIO_COMPLETE:** This indicates that ACIO loopback test using IOTE Engine A is complete
14:8	0h RO	**RESERVED0:** Reserved
7:0	0h RO	**MAX_COUNT:** 8-bit max count number when the test is done for IOTE ENGINE A

31.79 TX_DFX_PORT1_DELAY_CALIB_REG Register (TX_DFX_PORT1_DELAY_CALIB_REG)—Offset 138h

Delay Calibration Register

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RO	**RESERVED0:** Reserved
4	0h RO	**COMP_STATA:** comp_stata
3:0	0h RO	**CAL_DELAY1:** cal_delay1

31.80 TX_DFX_PORT1_RXLPBK_DATA Register (TX_DFX_PORT1_RXLPBK_DATA)—Offset 13Ch

RX Loopback Data Register

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RX_LPBK_DATA1:** rx_lpbk_data1 (CLK A) --> Use SIPO instead of Receiver data

31.81 TX_DFX_PORT1_CLANE_CTL_REG Register (TX_DFX_PORT1_CLANE_CTL_REG)—Offset 140h

TX DFX Clock Lane Control Reg

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RW	**RESERVED0:** Reserved
6:3	0h RW	**TIME_SKEW_ADJUST_CLK:** Lane Timing Skew adjustability between CLK/DATA for DSI with 4-bit control for each lane: ~25ps resolution
2	0h RW	**HS_TX_REGEN_CLK:** DFX override for HS TX REG enable
1	0h RW	**HS_TX_EN_CLK:** DFX override for HS TX enable
0	0h RW	**LP_TX_EN_CLK:** DFX override for LP TX enable

31.82 TX_DFX_PORT1_RESERVED1 Register (TX_DFX_PORT1_RESERVED1)—Offset 144h

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.83 TX_DFX_PORT1_RESERVED2 Register (TX_DFX_PORT1_RESERVED2)—Offset 148h

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.84 TX_DFX_PORT1_RESERVED3 Register (TX_DFX_PORT1_RESERVED3)—Offset 14Ch

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.85 TX_DFX_PORT1_DLANE0_CTL_REG Register (TX_DFX_PORT1_DLANE0_CTL_REG)—Offset 150h

TX DFX Data Lane-0 Control Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RW	**RESERVED0:** Reserved
17	0h RW	**LP_RX_EN:** LP RX DATA Lane-0 CNTL TESTMODE. DFX override for LP RX CNTL
16:9	0h RW	**COMP_SEL_OVRD:** (DSI_PIPEA_SIPO0) compare_sel_ovrd: Overriden value for symbol
8	0h RW	**CP_STAT:** (DSI_PIPEA_SIPO0) cp_stat: Enable symbol value from sync_align logic output
7	0h RW	**CP_STAT_OVRD:** (DSI_PIPEA_SIPO0) cp_stat_ovrd: 1 - select overriden value, 0 - select value from sync_align logic
6:3	0h RW	**TIME_SKEW_ADJUST_DATA:** Lane 0-3 Timing Skew adjustability between CLK/DATA for DSI with 4-bit control for each lane: ~25ps resolution
2	0h RW	**HS_TX_REGEN:** DFX override for HS TX DATA0 REG enable
1	0h RW	**HS_TX_EN:** DFX override for HS TX DATA0 enable
0	0h RW	**LP_TX_EN:** DFX override for LP TX DATA0 enable

31.86 TX_DFX_PORT1_DLANE0_STAT_REG Register (TX_DFX_PORT1_DLANE0_STAT_REG)—Offset 154h

TX DFX Data Lane-0 Status Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**RESERVED0:** Reserved
7:0	0h RO	**INTLPBK_COMP_STAT:** Compare status from SIPO0 - Pipe A

31.87 TX_DFX_PORT1_DLANE1_CTL_REG Register (TX_DFX_PORT1_DLANE1_CTL_REG)—Offset 158h

TX DFX Data Lane-1 Control Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RW	**RESERVED1:** Reserved
17	0h RW	**LP_RX_EN:** LP RX DATA Lane-1 CNTL TESTMODE. DFX override for LP RX CNTL
16:9	0h RW	**COMP_SEL_OVRD:** (DSI_PIPEA_SIPO0) compare_sel_ovrd: Overriden value for symbol
8	0h RW	**CP_STAT:** (DSI_PIPEA_SIPO0) cp_stat: Enable symbol value from sync_align logic output
7	0h RW	**CP_STAT_OVRD:** (DSI_PIPEA_SIPO0) cp_stat_ovrd: 1 - select overriden value, 0 - select value from sync_align logic
6:3	0h RW	**RESERVED0:** RESERVED
2	0h RW	**HS_TX_REGEN:** DFX override for HS TX DATA1 REG enable
1	0h RW	**HS_TX_EN:** DFX override for HS TX DATA1 enable
0	0h RW	**LP_TX_EN:** DFX override for LP TX DATA1 enable

31.88 TX_DFX_PORT1_DLANE1_STAT_REG Register (TX_DFX_PORT1_DLANE1_STAT_REG)—Offset 15Ch

TX DFX Data Lane-1 Status Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**RESERVED0:** Reserved
7:0	0h RO	**INTLPBK_COMP_STAT:** Compare status from SIPO1 - Pipe A

31.89 TX_DFX_PORT1_DLANE2_CTL_REG Register (TX_DFX_PORT1_DLANE2_CTL_REG)—Offset 160h

TX DFX Data Lane-2 Control Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RW	**RESERVED1:** Reserved
17	0h RW	**LP_RX_EN:** LP RX DATA Lane-2 CNTL TESTMODE. DFX override for LP RX CNTL
16:9	0h RW	**COMP_SEL_OVRD:** (DSI_PIPEA_SIPO0) compare_sel_ovrd: Overriden value for symbol
8	0h RW	**CP_STAT:** (DSI_PIPEA_SIPO0) cp_stat: Enable symbol value from sync_align logic output
7	0h RW	**CP_STAT_OVRD:** (DSI_PIPEA_SIPO0) cp_stat_ovrd: 1 - select overriden value, 0 - select value from sync_align logic
6:3	0h RW	**RESERVED0:** RESERVED
2	0h RW	**HS_TX_REGEN:** DFX override for HS TX DATA2REG enable
1	0h RW	**HS_TX_EN:** DFX override for HS TX DATA2enable
0	0h RW	**LP_TX_EN:** DFX override for LP TX DATA2enable

31.90 TX_DFX_PORT1_DLANE2_STAT_REG Register (TX_DFX_PORT1_DLANE2_STAT_REG)—Offset 164h

TX DFX Data Lane-2 Status Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**RESERVED0:** Reserved
7:0	0h RO	**INTLPBK_COMP_STAT:** Compare status from SIPO2 - Pipe A

31.91 TX_DFX_PORT1_DLANE3_CTL_REG Register (TX_DFX_PORT1_DLANE3_CTL_REG)—Offset 168h

TX DFX Data Lane-3 Control Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:18	0h RW	**RESERVED1:** Reserved
17	0h RW	**LP_RX_EN:** LP RX DATA Lane-3 CNTL TESTMODE. DFX override for LP RX CNTL
16:9	0h RW	**COMP_SEL_OVRD:** (DSI_PIPEA_SIPO0) compare_sel_ovrd: Overriden value for symbol
8	0h RW	**CP_STAT:** (DSI_PIPEA_SIPO0) cp_stat: Enable symbol value from sync_align logic output
7	0h RW	**CP_STAT_OVRD:** (DSI_PIPEA_SIPO0) cp_stat_ovrd: 1 - select overriden value, 0 - select value from sync_align logic
6:3	0h RW	**RESERVED0:** RESERVED
2	0h RW	**HS_TX_REGEN:** DFX override for HS TX DATA3REG enable
1	0h RW	**HS_TX_EN:** DFX override for HS TX DATA3enable
0	0h RW	**LP_TX_EN:** DFX override for LP TX DATA3enable

31.92 TX_DFX_PORT1_DLANE3_STAT_REG Register (TX_DFX_PORT1_DLANE3_STAT_REG)—Offset 16Ch

TX DFX Data Lane-3 Status Reg

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**RESERVED0:** Reserved
7:0	0h RO	**INTLPBK_COMP_STAT:** Compare status from SIPO3 - Pipe A

31.93 TX_DFX_PORT1_LANE_RESERVED5 Register (TX_DFX_PORT1_LANE_RESERVED5)—Offset 170h

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.94 TX_DFX_PORT1_LANE_RESERVED6 Register (TX_DFX_PORT1_LANE_RESERVED6)—Offset 174h

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.95 TX_DFX_PORT1_LANE_RESERVED7 Register (TX_DFX_PORT1_LANE_RESERVED7)—Offset 178h

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.96 TX_DFX_PORT1_LANE_RESERVED8 Register (TX_DFX_PORT1_LANE_RESERVED8)—Offset 17Ch

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.97 TX_DFE_PORT1_DLN_CNT_LPX Register (TX_DFE_PORT1_DLN_CNT_LPX)—Offset 180h

lpx count register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: Ah

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	Ah RW	**DLN_CNT_LPX:** period for which a LP state should be driven

31.98 TX_DFE_PORT1_DLN_CNT_HS_PREP Register (TX_DFE_PORT1_DLN_CNT_HS_PREP)—Offset 184h

hs prepare count register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 9h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	9h RW	**DLN_CNT_HS_PREP:** Period for which HS Prepare time should be accomodated

31.99 TX_DFE_PORT1_DLN_CNT_HS_ZERO Register (TX_DFE_PORT1_DLN_CNT_HS_ZERO)—Offset 188h

hs zero count register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 15h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	15h RW	**DLN_CNT_HS_ZERO:** Period for which HS Zero should be driven before sync

31.100 TX_DFE_PORT1_DLN_CNT_HS_TRAIL Register (TX_DFE_PORT1_DLN_CNT_HS_TRAIL)—Offset 18Ch

hs trail count register

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: Bh

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	Bh RW	**DLN_CNT_HS_TRAIL:** Period for which HS trailing should be driven

31.101 TX_DFE_PORT1_DLN_CNT_HS_EXIT Register (TX_DFE_PORT1_DLN_CNT_HS_EXIT)—Offset 190h

hs exit count register

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 12h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	12h RW	**DLN_CNT_HS_EXIT:** Period for which HS exit state should be maintained

31.102 TX_DFE_PORT1_DLN_CNT_RX Register (TX_DFE_PORT1_DLN_CNT_RX)—Offset 194h

rx count register

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	1h RW	**DLN_CNT_RX:** Counter that controls the assertion of enable on the DPHY

31.103 TX_DFE_PORT1_DLN_SYNC_CNT Register (TX_DFE_PORT1_DLN_SYNC_CNT)—Offset 198h

sync count register

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 1h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	1h RW	**DLN_SYNC_COUNT:** A timeout value used for Sync error detector logic

31.104 TX_DFE_PORT1_CLN_CNT_HS_TRAIL Register (TX_DFE_PORT1_CLN_CNT_HS_TRAIL)—Offset 19Ch

clane trail count

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: Bh

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	Bh RW	**CLN_HS_TRAIL:** Wait time in byte clock for the trailing bits

31.105 TX_DFE_PORT1_CLN_CNT_HS_EXIT Register (TX_DFE_PORT1_CLN_CNT_HS_EXIT)—Offset 1A0h

clane exit count

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 12h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	12h RW	**CLN_HS_EXIT:** Wait time in byte clock for the exit state

31.106 TX_DFE_PORT1_CLN_CNT_LPX Register (TX_DFE_PORT1_CLN_CNT_LPX)—Offset 1A4h

clane lpx count

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 9h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	9h RW	**CLN_CNT_LPX:** Wait time in byte clock for the LPX

31.107 TX_DFE_PORT1_CLN_CNT_PREP Register (TX_DFE_PORT1_CLN_CNT_PREP)—Offset 1A8h

clane prepare count

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: Eh

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	Eh RW	**CLN_CNT_PREP:** Wait time in byte clock for the prepare time

31.108 TX_DFE_PORT1_CLN_CNT_ZERO Register (TX_DFE_PORT1_CLN_CNT_ZERO)—Offset 1ACh

clane zero count

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 32h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	32h RW	**CLN_CNT_ZERO:** Wait time in byte clock for the zero state

31.109 TX_DFE_PORT1_CLN_CNT_PLL Register (TX_DFE_PORT1_CLN_CNT_PLL)—Offset 1B0h

clane PLL count

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**RESERVED0:** Reserved
15:0	0h RW	**CLN_CNT_PLL:** Count value used for the PLL lock time

31.110 TX_DFE_PORT1_CLN_CNT_POST Register (TX_DFE_PORT1_CLN_CNT_POST)—Offset 1B4h

clane cnt post

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 12h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** Reserved
7:0	12h RW	**CLN_CNT_POST:** Wait time in byte clock for the zero state

31.111 TX_DFE_PORT1_CLN_DLN_ENABLE_OVR Register (TX_DFE_PORT1_CLN_DLN_ENABLE_OVR)—Offset 1B8h

Clock Lane and Data Lane Enable Override

Access Method

Type: MSG Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**RESERVED0:** RESERVED
7	0h RW	**AFE_LPRX_EN_CNTRL_SEL:** AFE Low Power Receiver Enable Control Select on the functional Path (i.e before the DFX override) 1 : LP Receiver Enable for Lane 0 is always Enable 0 : Use Data Lane Enables to qualify LP Receiver Enable Control
6	0h RW	**CLN_ENABLE_OVR_SEL:** TX-DFE Clock Lane Enable Override Select Control 1 : Override the Clock Lane Enable from this reg (CLN_ENABL_OVR) 0 : Select the Clock Lane Enable from ORing Data Lane Enable Controls After Override Mux
5:2	0h RW	**DLN_ENABLE_OVR:** TX-DFE Data Lane Enable Control Override Signal DLN_ENABLE_OVR[3:0] DLN_ENABLE_OVR[3] : 1 = Enable the Data Lane-3 ; 0 = Disable Data Lane-3 DLN_ENABLE_OVR[2] : 1 = Enable the Data Lane-2 ; 0 = Disable Data Lane-2 DLN_ENABLE_OVR[1] : 1 = Enable the Data Lane-1 ; 0 = Disable Data Lane-1 DLN_ENABLE_OVR[0] : 1 = Enable the Data Lane-0 ; 0 = Disable Data Lane-0
1	0h RW	**CLN_ENABLE_OVR:** TX-DFE Clock Lane Enable Control Override Signal 1 : Enable the Clock Lane 0 : Disable the Clock Lane [Reset Default]
0	0h RW	**DLN_ENABLE_OVR_SEL:** TX-DFE Data Lane Enable Override Select Control 1 : Override the Data Lane Enable from this reg 0 : Select the Data Lane Enable from PPI or normal TX-DFE control [reset Default]

31.112 TX_DFE_PORT1_RESERVED1 Register (TX_DFE_PORT1_RESERVED1)—Offset 1BCh

RESERVED

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**RESERVED0:** Reserved for future enhancement

31.113 RX_DWORD5 (rx_dword5)—Offset 4214h/4614h/1A14h/1614h/1214h/614h/614h/1E14h/4E14h/4A14h

RX_DWORD5

Access Method

Type: IO Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7:5	0h RW	**ocfgbonus3_sus_2_0 (ocfgbonus3_sus_2_0):** Bonus suswell register
4:0	0h RO	Reserved.

§ §

32 Display Registers

32.1 LJPLL_RW_CONTROL_0 (LJPLL_CR_RW_CONTROL_0)—Offset 0h

LJPLL CR

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 41h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RW	**RESERVED_0**
25	0h RW	**SKIP_FUSE_PULL:** If set Don t repull fuses on side_rst deassertion
24:18	0h RW	**LJPLL_OUTPUT_RATIO:** Set the output clk ratio
17:16	0h RW	**LJPLL_PVD_RATIO:** Set the post VCO clk divider ratio 001 012 104 118
15:14	0h RW	**LJPLL_REF_RATIO:** Set the refclk ratio 00 refclk 01 refclk/2 10 refclk/4 11refclk/8 set this to 2 b10 for Iunit PLL
13	0h RW	**LJPLL_FORCE_ON:** Force The PLL Enable ON 0NoForce 1 FORCE_ON
12	0h RW	**LJPLL_FORCE_OFF:** Force The PLL Enable OFF 0 NoForce 1 FORCE_OFF
11:10	0h RW	**SEL_MIPICLK_C:** Select DSI Clk 00 default 018x 1016x/3 114x don t care for other PLLs
9:8	0h RW	**SEL_MIPICLK_A:** Select DSI Clk 00default 018x 1016x/3 114x don t care for other PLLs
7:0	41h RW	**LJPLL_FB_RATIO:** Set the feedback ratio

32.2 LJPLL_RW_CONTROL_1 (LJPLL_CR_RW_CONTROL_1)—Offset 10h

LJPLL CR

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**PLL_RATIO_FRAC:** Clock Bending: fractional frequency multiplier; shift PLL clock frequency by (value/2^24)*refclk frequency. eg 0x200000 = (2097152/2^24) * refclk freq = 0.125*19.2 = 2.4MHz
7:2	0h RW	**SPARE:** Spare CR
1	0h RW	**SSC_EN_OVRD:** Override the fuse/tap value for SSC enable
0	0h RW	**SSC_EN:** Spread Spectrum Clocking: spread enable; 0x0=no frequency spreading; 0x1=enable frequency spreading on PLL output clock; This enable is not default and needs to be set along with SSC_EN_OVRD to take effect

32.3 LJPLL_RW_CONTROL_2 (LJPLL_CR_RW_CONTROL_2)—Offset 14h

LJPLL CR

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**SSC_FRAC_STEP:** Spread Spectrum Clocking: fractional step configuration; fraction of PLL ratio at which to take frequency modulation steps. eg 0x200000 = (2097152/2^20) * refclk freq = 0.125*19.2 = 2.4MHz steps. Spread magnitude is determined by the step size multiplied by the number of steps in the modulation period (see ssc_cyc_to_peak_m1 for steps per modulation period).
11	0h RW	**SPARE:** Spare CR
10:9	0h RW	**SSC_MODE:** Spread Spectrum Clocking: spread direction select; 0x0 = down-spread only; 0x1 = up-spread only; 0x2 = center spread, start with down-spread; 0x3 = center spread, start with up-spread
8:0	0h RW	**SSC_CYC_TO_PEAK_M1:** Spread Spectrum Clocking: spread period configuration; half the number of steps in the modulation period minus 1. Period of modulation is 2*(value+1) multiplied by the step duration (PLL refclk period). eg 0x12B = 2*(299+1) * (1/19.2MHz) = 600 * 52.083ns = 31.25us. Spread magnitude is determined by the step size (integer + fractional) multiplied by the number of steps in the modulation period (see ssc_frac_step and ssc_ratio_step for step

32.4 depll_cp (depll_cp)—Offset 20h

Policy DEPLL_POLICY_GROUP CP Register

Access Method

Type: MEM Register
(Size: 64 bits)

Device:
Function:

Default: 40001200215h

Bit Range	Default & Access	Field Name (ID): Description
63:0	40001200215h RW	**sai**

32.5 depll_rac (depll_rac)—Offset 28h

Policy DEPLL_POLICY_GROUP RAC Register

Access Method

Type: MEM Register (Size: 64 bits)
Device:
Function:

Default: 40001200215h

Bit Range	Default & Access	Field Name (ID): Description
63:0	40001200215h RW	**sai**

32.6 depll_wac (depll_wac)—Offset 30h

Policy DEPLL_POLICY_GROUP WAC Register

Access Method

Type: MEM Register (Size: 64 bits)
Device:
Function:

Default: 40001200215h

Bit Range	Default & Access	Field Name (ID): Description
63:0	40001200215h RW	**sai**

32.7 MCHBAR_LSB (GFXVTDBAR_LSB)—Offset 6C88h

This is the base address for the Graphics VT configuration space. There is no physical memory within this 4KB window that can be addressed. The 4KB reserved by this register does not alias to any PCI 2.3 compliant memory mapped space. On reset, the GFX-VT configuration space is disabled and must be enabled by writing a 1 to GFX-VTBAREN. All the bits in this register are locked in Intel TXT mode. BIOS programs this register after which the register cannot be altered. This register is only written via an SAI protected IOSF SB shadow access.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**GFXVTBAR (GFXVTBAR):** This field corresponds to bits 38 to 12 of the base address GFX-VT configuration space. BIOS will program this register resulting in a base address for a 4KB block of contiguous memory address space. This register ensures that a naturally aligned 4KB space is allocated within the first 512GB of addressable memory space. System Software uses this base address to program the GFX-VT register set. All the Bits in this register are locked in Intel TXT mode.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**SPARE (SPARE):** Was lock bit prior to Gen10
0	0h RW/V	**GFXVTBAREN (GFXVTBAREN):** 0: GFX-VTBAR is disabled and does not claim any memory. 1: GFX-VTBAR memory mapped accesses are claimed and decoded appropriately This bit will remain 0 if VTd capability is disabled.

32.8 MCHBAR_MSB (GFXVTDBAR_MSB)—Offset 6C8Ch

This is the base address for the Graphics VT configuration space. There is no physical memory within this 4KB window that can be addressed. The 4KB reserved by this register does not alias to any PCI 2.3 compliant memory mapped space. On reset, the GFX-VT configuration space is disabled and must be enabled by writing a 1 to GFX-VTBAREN. All the bits in this register are locked in Intel TXT mode. BIOS programs this register after which the register cannot be altered. This register is only written via an SAI protected IOSF SB shadow access.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GFXVTBAR (GFXVTBAR):** This field corresponds to bits63 to 12 of the base address GFX-VT configuration space. BIOS will program this register resulting in a base address for a 4KB block of contiguous memory address space. This register ensures that a naturally aligned 4KB space is allocated within the first 512GB of addressable memory space. System Software uses this base address to program the GFX-VT register set. All the Bits in this register are locked in Intel TXT mode.

32.9 Fuse Status (FUSE_STATUS)—Offset 42000h

This register is on the ungated clock and the chip reset, not the FLR or display debug reset.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**FUSE DOWNLOAD STATUS (Fuse_Download_Status):** This field indicates the status of fuse and strap download to the Display Engine. After fuse and strap download, fuses will be distributed within the Display Engine.
30:28	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'Fuse Status' does not have a description in the BXML
27	0h RO	**FUSE PG0 DISTRIBUTION STATUS (Fuse_PG0_Distribution_Status):** This field indicates the status of fuse distribution to power well #0.
26	0h RO	**FUSE PG1 DISTRIBUTION STATUS (Fuse_PG1_Distribution_Status):** This field indicates the status of fuse distribution to power well #1.
25	0h RO	**FUSE PG2 DISTRIBUTION STATUS (Fuse_PG2_Distribution_Status):** This field indicates the status of fuse distribution to power well #2.
24:0	0h RSV	**RESERVED (Reserved_1):** The field 'Reserved' in register 'Fuse Status' does not have a description in the BXML

32.10 Display Engine Power 1 (DE_POWER1)—Offset 42400h

The register 'Display Engine Power 1' does not have a description in the BXML source

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**POWER WELL 2 STATE (Power_Well_2_State):** This field indicates the status of display power well 2.
30	0h RO	**DISPLAY PIPES ENABLED (Display_Pipes_Enabled):** This field indicates if any display pipes are enabled.
29	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'Display Engine Power 1' does not have a description in the BXML
28	0h RO	**POWER WELL 1 STATE (Power_Well_1_State):** This field indicates the status of display power well 1.
27:26	0h RO	**SRD STATUS (SRD_Status):** This field indicates the live status of the SRD link on eDP DDI-A.
25	0h RO	**KVM SESSION STATUS (KVM_Session_Status):** This field indicates the status of KVM session.
24:13	0h RSV	**RESERVED (Reserved_1):** The field 'Reserved' in register 'Display Engine Power 1' does not have a description in the BXML
12:10	0h RO	**ENABLED PIPE SCALERS (Enabled_Pipe_Scalers):** Indicates total usage of the EBBs.

Bit Range	Default & Access	Field Name (ID): Description
9:8	0h RSV	**RESERVED (Reserved_2):** The field 'Reserved' in register 'Display Engine Power 1' does not have a description in the BXML
7:4	0h RO	**TRANSMIT LANES ENABLED (Transmit_Lanes_Enabled):** The total number of DDI lanes enabled.
3	0h RSV	**RESERVED (Reserved_3):** The field 'Reserved' in register 'Display Engine Power 1' does not have a description in the BXML
2:0	0h RO	**ENABLED DPLLS (Enabled_DPLLs):** The total number of Display PLLs enabled.

32.11 Display Engine Power 2 (DE_POWER2)—Offset 42404h

The register 'Display Engine Power 2' does not have a description in the BXML source

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO	**DE BANDWIDTH COUNTER (DE_bandwidth_counter):** This counter increments on every cache line put arriving at the DE. The bandwidth is estimated by taking the difference between two reads at a known interval. Access is actually a read/write variant. Writes to this register will load the write data into the counter. The GDR_DFT register Hold_DE_POWER2 bit can be set to stop the count from incrementing.

32.12 Master Interrupt Control (MASTER_INT_CTL)—Offset 44200h

This register has the master enable for graphics interrupts and gives an overview of what interrupts are pending. An interrupt pending bit will read 1b while one or more interrupts of that category are set (IIR) and enabled (IER). All Pending Interrupts are ORed together to generate the combined interrupt. The combined interrupt is ANDed with the Master Interrupt enable to create the master enabled interrupt. The master enabled interrupt goes to PCI device 2 interrupt processing. The master interrupt enable must be set before any of these interrupts will propagate to PCI device 2 interrupt processing.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**MASTER INTERRUPT ENABLE (Master_Interrupt_Enable):** This is the master control for graphics interrupts. This must be enabled for any of these interrupts to propagate to PCI device 2 interrupt processing.
30	0h RO	**PCU INTERRUPTS PENDING (PCU_Interrupts_Pending):** This field indicates if interrupts of this category are pending.
29:25	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'Master Interrupt Control' does not have a description in the BXML
24	0h RO	**AUDIO CODEC INTERRUPTS PENDING (Audio_Codec_Interrupts_Pending):** This field indicates if interrupts of this category are pending.
23	0h RO	**DE PCH INTERRUPTS PENDING (DE_PCH_Interrupts_Pending):** This field indicates if interrupts of this category are pending. The PCH Display interrupt is configured through the SDE interrupt registers.
22	0h RO	**DE MISC INTERRUPTS PENDING (DE_Misc_Interrupts_Pending):** This field indicates if interrupts of this category are pending.
21	0h RSV	**RESERVED (Reserved_1):** The field 'Reserved' in register 'Master Interrupt Control' does not have a description in the BXML
20	0h RO	**DE PORT INTERRUPTS PENDING (DE_Port_Interrupts_Pending):** This field indicates if interrupts of this category are pending.
19	0h RSV	**RESERVED (Reserved_2):** The field 'Reserved' in register 'Master Interrupt Control' does not have a description in the BXML
18	0h RO	**DE PIPE C INTERRUPTS PENDING (DE_Pipe_C_Interrupts_Pending):** This field indicates if interrupts of this category are pending.
17	0h RO	**DE PIPE B INTERRUPTS PENDING (DE_Pipe_B_Interrupts_Pending):** This field indicates if interrupts of this category are pending.
16	0h RO	**DE PIPE A INTERRUPTS PENDING (DE_Pipe_A_Interrupts_Pending):** This field indicates if interrupts of this category are pending.
15:8	0h RSV	**RESERVED (Reserved_3):** The field 'Reserved' in register 'Master Interrupt Control' does not have a description in the BXML
7	0h RO	**WDBOX OR OACS INTERRUPTS PENDING (WDBox_or_OACS_Interrupts_Pending):** This field indicates if interrupts of this category are pending.
6	0h RO	**VEBOX INTERRUPTS PENDING (VEBox_Interrupts_Pending):** This field indicates if interrupts of this category are pending.
5	0h RO	Reserved.
4	0h RO	**GTPM INTERRUPTS PENDING (GTPM_Interrupts_Pending):** This field indicates if interrupts of this category are pending.
3	0h RO	**VCS2 INTERRUPTS PENDING (VCS2_Interrupts_Pending):** This field indicates if interrupts of this category are pending.
2	0h RO	**VCS1 INTERRUPTS PENDING (VCS1_Interrupts_Pending):** This field indicates if interrupts of this category are pending.
1	0h RO	**BLITTER INTERRUPTS PENDING (Blitter_Interrupts_Pending):** This field indicates if interrupts of this category are pending.
0	0h RO	**RENDER INTERRUPTS PENDING (Render_Interrupts_Pending):** This field indicates if interrupts of this category are pending.

32.13 GT 0 Interrupts 0 (GT_0_INTERRUPT_0)—Offset 44300h

This table indicates which events are mapped to each bit of the GT Interrupt 0 registers. Bits 15:0 are used for Render CS. Bits 31:16 are used for Blitter CS. The IER enabled Render Interrupt IIR (sticky) bits are ORed together to generate the Render Interrupts Pending bit in the Master Interrupt Control register. The IER enabled Blitter Interrupt IIR (sticky) bits are ORed together to generate the Blitter Interrupts Pending bit in the Master Interrupt Control register. 0x44300 = ISR 0x44304 = IMR 0x44308 = IIR 0x4430C = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h	**SPARE 31 (Spare_31):** The field 'Spare 31' in register 'GT 0 Interrupts 0' does not have a description in the BXML
30	0h	**SPARE 30 (Spare_30):** The field 'Spare 30' in register 'GT 0 Interrupts 0' does not have a description in the BXML
29	0h	**SPARE 29 (Spare_29):** The field 'Spare 29' in register 'GT 0 Interrupts 0' does not have a description in the BXML
28	0h	**SPARE 28 (Spare_28):** The field 'Spare 28' in register 'GT 0 Interrupts 0' does not have a description in the BXML
27	0h	**BCS WAIT ON SEMAPHORE (BCS_Wait_On_Semaphore):** The field 'BCS Wait On Semaphore' in register 'GT 0 Interrupts 0' does not have a description in the BXML
26	0h	**SPARE 26 (Spare_26):** The field 'Spare 26' in register 'GT 0 Interrupts 0' does not have a description in the BXML
25	0h	**SPARE 25 (Spare_25):** The field 'Spare 25' in register 'GT 0 Interrupts 0' does not have a description in the BXML
24	0h	**BCS CONTEXT SWITCH INTERRUPT (BCS_Context_Switch_Interrupt):** The field 'BCS Context Switch Interrupt' in register 'GT 0 Interrupts 0' does not have a description in the BXML
23	0h	**SPARE 23 (Spare_23):** The field 'Spare 23' in register 'GT 0 Interrupts 0' does not have a description in the BXML
22	0h RO	Reserved.
21	0h	**SPARE 21 (Spare_21):** The field 'Spare 21' in register 'GT 0 Interrupts 0' does not have a description in the BXML
20	0h	**BCS MI FLUSH DW NOTIFY (BCS_MI_Flush_DW_Notify):** The field 'BCS MI Flush DW Notify' in register 'GT 0 Interrupts 0' does not have a description in the BXML
19	0h	**BCS ERROR INTERRUPT (BCS_Error_Interrupt):** The field 'BCS Error Interrupt' in register 'GT 0 Interrupts 0' does not have a description in the BXML
18	0h	**SPARE 18 (Spare_18):** The field 'Spare 18' in register 'GT 0 Interrupts 0' does not have a description in the BXML
17	0h	**SPARE 17 (Spare_17):** The field 'Spare 17' in register 'GT 0 Interrupts 0' does not have a description in the BXML
16	0h	**BCS MI USER INTERRUPT (BCS_MI_User_Interrupt):** The field 'BCS MI User Interrupt' in register 'GT 0 Interrupts 0' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
15	0h	**SPARE 15 (Spare_15):** The field 'Spare 15' in register 'GT 0 Interrupts 0' does not have a description in the BXML
14	0h	**SPARE 14 (Spare_14):** The field 'Spare 14' in register 'GT 0 Interrupts 0' does not have a description in the BXML
13	0h	**SPARE 13 (Spare_13):** The field 'Spare 13' in register 'GT 0 Interrupts 0' does not have a description in the BXML
12	0h	**SPARE 12 (Spare_12):** The field 'Spare 12' in register 'GT 0 Interrupts 0' does not have a description in the BXML
11	0h	**CS WAIT ON SEMAPHORE (CS_Wait_On_Semaphore):** The field 'CS Wait On Semaphore' in register 'GT 0 Interrupts 0' does not have a description in the BXML
10	0h	**CS L3 COUNTER SAVE (CS_L3_Counter_Save):** The field 'CS L3 Counter Save' in register 'GT 0 Interrupts 0' does not have a description in the BXML
9	0h	**CS TR INVALID TILE DETECTION (CS_TR_Invalid_Tile_Detection):** The field 'CS TR Invalid Tile Detection' in register 'GT 0 Interrupts 0' does not have a description in the BXML
8	0h	**CS CONTEXT SWITCH INTERRUPT (CS_Context_Switch_Interrupt):** The field 'CS Context Switch Interrupt' in register 'GT 0 Interrupts 0' does not have a description in the BXML
7	0h	**PAGE FAULT INTERRUPT (Page_Fault_Interrupt):** This interrupt is for handling Legacy Page Fault interface for all Command Streamers [BCS, RCS, VCS, VECS]. When Fault Repair Mode is enabled, Interrupt mask register value is not looked at to generate interrupt due to page fault. Please refer to vol1c 'page fault support' section for more details. In Advanced (PRQ) Fault Interface is done through GUC interface.
6	0h	**CS WATCHDOG COUNTER EXPIRED (CS_Watchdog_Counter_Expired):** The field 'CS Watchdog Counter Expired' in register 'GT 0 Interrupts 0' does not have a description in the BXML
5	0h	**SPARE 5 (Spare_5):** The field 'Spare 5' in register 'GT 0 Interrupts 0' does not have a description in the BXML
4	0h	**CS PIPE_CONTROL NOTIFY (CS_PIPE_CONTROL_Notify):** The field 'CS PIPE_CONTROL Notify' in register 'GT 0 Interrupts 0' does not have a description in the BXML
3	0h	**CS ERROR INTERRUPT (CS_Error_Interrupt):** The field 'CS Error Interrupt' in register 'GT 0 Interrupts 0' does not have a description in the BXML
2	0h	**SPARE 2 (Spare_2):** The field 'Spare 2' in register 'GT 0 Interrupts 0' does not have a description in the BXML
1	0h RO	Reserved.
0	0h	**CS MI USER INTERRUPT (CS_MI_User_Interrupt):** The field 'CS MI User Interrupt' in register 'GT 0 Interrupts 0' does not have a description in the BXML

32.14 GT 0 Interrupts 1 (GT_0_INTERRUPT_1)—Offset 44304h

This table indicates which events are mapped to each bit of the GT Interrupt 0 registers. Bits 15:0 are used for Render CS. Bits 31:16 are used for Blitter CS. The IER enabled Render Interrupt IIR (sticky) bits are ORed together to generate the Render Interrupts Pending bit in the Master Interrupt Control register. The IER enabled Blitter Interrupt IIR (sticky) bits are ORed together to generate the Blitter Interrupts Pending bit in the Master Interrupt Control register. 0x44300 = ISR 0x44304 = IMR 0x44308 = IIR 0x4430C = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h	**SPARE 31 (Spare_31):** The field 'Spare 31' in register 'GT 0 Interrupts 1' does not have a description in the BXML
30	0h	**SPARE 30 (Spare_30):** The field 'Spare 30' in register 'GT 0 Interrupts 1' does not have a description in the BXML
29	0h	**SPARE 29 (Spare_29):** The field 'Spare 29' in register 'GT 0 Interrupts 1' does not have a description in the BXML
28	0h	**SPARE 28 (Spare_28):** The field 'Spare 28' in register 'GT 0 Interrupts 1' does not have a description in the BXML
27	0h	**BCS WAIT ON SEMAPHORE (BCS_Wait_On_Semaphore):** The field 'BCS Wait On Semaphore' in register 'GT 0 Interrupts 1' does not have a description in the BXML
26	0h	**SPARE 26 (Spare_26):** The field 'Spare 26' in register 'GT 0 Interrupts 1' does not have a description in the BXML
25	0h	**SPARE 25 (Spare_25):** The field 'Spare 25' in register 'GT 0 Interrupts 1' does not have a description in the BXML
24	0h	**BCS CONTEXT SWITCH INTERRUPT (BCS_Context_Switch_Interrupt):** The field 'BCS Context Switch Interrupt' in register 'GT 0 Interrupts 1' does not have a description in the BXML
23	0h	**SPARE 23 (Spare_23):** The field 'Spare 23' in register 'GT 0 Interrupts 1' does not have a description in the BXML
22	0h RO	Reserved.
21	0h	**SPARE 21 (Spare_21):** The field 'Spare 21' in register 'GT 0 Interrupts 1' does not have a description in the BXML
20	0h	**BCS MI FLUSH DW NOTIFY (BCS_MI_Flush_DW_Notify):** The field 'BCS MI Flush DW Notify' in register 'GT 0 Interrupts 1' does not have a description in the BXML
19	0h	**BCS ERROR INTERRUPT (BCS_Error_Interrupt):** The field 'BCS Error Interrupt' in register 'GT 0 Interrupts 1' does not have a description in the BXML
18	0h	**SPARE 18 (Spare_18):** The field 'Spare 18' in register 'GT 0 Interrupts 1' does not have a description in the BXML
17	0h	**SPARE 17 (Spare_17):** The field 'Spare 17' in register 'GT 0 Interrupts 1' does not have a description in the BXML
16	0h	**BCS MI USER INTERRUPT (BCS_MI_User_Interrupt):** The field 'BCS MI User Interrupt' in register 'GT 0 Interrupts 1' does not have a description in the BXML
15	0h	**SPARE 15 (Spare_15):** The field 'Spare 15' in register 'GT 0 Interrupts 1' does not have a description in the BXML
14	0h	**SPARE 14 (Spare_14):** The field 'Spare 14' in register 'GT 0 Interrupts 1' does not have a description in the BXML
13	0h	**SPARE 13 (Spare_13):** The field 'Spare 13' in register 'GT 0 Interrupts 1' does not have a description in the BXML
12	0h	**SPARE 12 (Spare_12):** The field 'Spare 12' in register 'GT 0 Interrupts 1' does not have a description in the BXML
11	0h	**CS WAIT ON SEMAPHORE (CS_Wait_On_Semaphore):** The field 'CS Wait On Semaphore' in register 'GT 0 Interrupts 1' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
10	0h	**CS L3 COUNTER SAVE (CS_L3_Counter_Save):** The field 'CS L3 Counter Save' in register 'GT 0 Interrupts 1' does not have a description in the BXML
9	0h	**CS TR INVALID TILE DETECTION (CS_TR_Invalid_Tile_Detection):** The field 'CS TR Invalid Tile Detection' in register 'GT 0 Interrupts 1' does not have a description in the BXML
8	0h	**CS CONTEXT SWITCH INTERRUPT (CS_Context_Switch_Interrupt):** The field 'CS Context Switch Interrupt' in register 'GT 0 Interrupts 1' does not have a description in the BXML
7	0h	**PAGE FAULT INTERRUPT (Page_Fault_Interrupt):** This interrupt is for handling Legacy Page Fault interface for all Command Streamers [BCS, RCS, VCS, VECS]. When Fault Repair Mode is enabled, Interrupt mask register value is not looked at to generate interrupt due to page fault. Please refer to vol1c 'page fault support' section for more details. In Advanced (PRQ) Fault Interface is done through GUC interface.
6	0h	**CS WATCHDOG COUNTER EXPIRED (CS_Watchdog_Counter_Expired):** The field 'CS Watchdog Counter Expired' in register 'GT 0 Interrupts 1' does not have a description in the BXML
5	0h	**SPARE 5 (Spare_5):** The field 'Spare 5' in register 'GT 0 Interrupts 1' does not have a description in the BXML
4	0h	**CS PIPE_CONTROL NOTIFY (CS_PIPE_CONTROL_Notify):** The field 'CS PIPE_CONTROL Notify' in register 'GT 0 Interrupts 1' does not have a description in the BXML
3	0h	**CS ERROR INTERRUPT (CS_Error_Interrupt):** The field 'CS Error Interrupt' in register 'GT 0 Interrupts 1' does not have a description in the BXML
2	0h	**SPARE 2 (Spare_2):** The field 'Spare 2' in register 'GT 0 Interrupts 1' does not have a description in the BXML
1	0h RO	Reserved.
0	0h	**CS MI USER INTERRUPT (CS_MI_User_Interrupt):** The field 'CS MI User Interrupt' in register 'GT 0 Interrupts 1' does not have a description in the BXML

32.15 GT 0 Interrupts 2 (GT_0_INTERRUPT_2)—Offset 44308h

This table indicates which events are mapped to each bit of the GT Interrupt 0 registers. Bits 15:0 are used for Render CS. Bits 31:16 are used for Blitter CS. The IER enabled Render Interrupt IIR (sticky) bits are ORed together to generate the Render Interrupts Pending bit in the Master Interrupt Control register. The IER enabled Blitter Interrupt IIR (sticky) bits are ORed together to generate the Blitter Interrupts Pending bit in the Master Interrupt Control register. 0x44300 = ISR 0x44304 = IMR 0x44308 = IIR 0x4430C = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h	**SPARE 31 (Spare_31):** The field 'Spare 31' in register 'GT 0 Interrupts 2' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
30	0h	**SPARE 30 (Spare_30):** The field 'Spare 30' in register 'GT 0 Interrupts 2' does not have a description in the BXML
29	0h	**SPARE 29 (Spare_29):** The field 'Spare 29' in register 'GT 0 Interrupts 2' does not have a description in the BXML
28	0h	**SPARE 28 (Spare_28):** The field 'Spare 28' in register 'GT 0 Interrupts 2' does not have a description in the BXML
27	0h	**BCS WAIT ON SEMAPHORE (BCS_Wait_On_Semaphore):** The field 'BCS Wait On Semaphore' in register 'GT 0 Interrupts 2' does not have a description in the BXML
26	0h	**SPARE 26 (Spare_26):** The field 'Spare 26' in register 'GT 0 Interrupts 2' does not have a description in the BXML
25	0h	**SPARE 25 (Spare_25):** The field 'Spare 25' in register 'GT 0 Interrupts 2' does not have a description in the BXML
24	0h	**BCS CONTEXT SWITCH INTERRUPT (BCS_Context_Switch_Interrupt):** The field 'BCS Context Switch Interrupt' in register 'GT 0 Interrupts 2' does not have a description in the BXML
23	0h	**SPARE 23 (Spare_23):** The field 'Spare 23' in register 'GT 0 Interrupts 2' does not have a description in the BXML
22	0h RO	Reserved.
21	0h	**SPARE 21 (Spare_21):** The field 'Spare 21' in register 'GT 0 Interrupts 2' does not have a description in the BXML
20	0h	**BCS MI FLUSH DW NOTIFY (BCS_MI_Flush_DW_Notify):** The field 'BCS MI Flush DW Notify' in register 'GT 0 Interrupts 2' does not have a description in the BXML
19	0h	**BCS ERROR INTERRUPT (BCS_Error_Interrupt):** The field 'BCS Error Interrupt' in register 'GT 0 Interrupts 2' does not have a description in the BXML
18	0h	**SPARE 18 (Spare_18):** The field 'Spare 18' in register 'GT 0 Interrupts 2' does not have a description in the BXML
17	0h	**SPARE 17 (Spare_17):** The field 'Spare 17' in register 'GT 0 Interrupts 2' does not have a description in the BXML
16	0h	**BCS MI USER INTERRUPT (BCS_MI_User_Interrupt):** The field 'BCS MI User Interrupt' in register 'GT 0 Interrupts 2' does not have a description in the BXML
15	0h	**SPARE 15 (Spare_15):** The field 'Spare 15' in register 'GT 0 Interrupts 2' does not have a description in the BXML
14	0h	**SPARE 14 (Spare_14):** The field 'Spare 14' in register 'GT 0 Interrupts 2' does not have a description in the BXML
13	0h	**SPARE 13 (Spare_13):** The field 'Spare 13' in register 'GT 0 Interrupts 2' does not have a description in the BXML
12	0h	**SPARE 12 (Spare_12):** The field 'Spare 12' in register 'GT 0 Interrupts 2' does not have a description in the BXML
11	0h	**CS WAIT ON SEMAPHORE (CS_Wait_On_Semaphore):** The field 'CS Wait On Semaphore' in register 'GT 0 Interrupts 2' does not have a description in the BXML
10	0h	**CS L3 COUNTER SAVE (CS_L3_Counter_Save):** The field 'CS L3 Counter Save' in register 'GT 0 Interrupts 2' does not have a description in the BXML
9	0h	**CS TR INVALID TILE DETECTION (CS_TR_Invalid_Tile_Detection):** The field 'CS TR Invalid Tile Detection' in register 'GT 0 Interrupts 2' does not have a description in the BXML
8	0h	**CS CONTEXT SWITCH INTERRUPT (CS_Context_Switch_Interrupt):** The field 'CS Context Switch Interrupt' in register 'GT 0 Interrupts 2' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
7	0h	**PAGE FAULT INTERRUPT (Page_Fault_Interrupt):** This interrupt is for handling Legacy Page Fault interface for all Command Streamers [BCS, RCS, VCS, VECS]. When Fault Repair Mode is enabled, Interrupt mask register value is not looked at to generate interrupt due to page fault. Please refer to vol1c 'page fault support' section for more details. In Advanced (PRQ) Fault Interface is done through GUC interface.
6	0h	**CS WATCHDOG COUNTER EXPIRED (CS_Watchdog_Counter_Expired):** The field 'CS Watchdog Counter Expired' in register 'GT 0 Interrupts 2' does not have a description in the BXML
5	0h	**SPARE 5 (Spare_5):** The field 'Spare 5' in register 'GT 0 Interrupts 2' does not have a description in the BXML
4	0h	**CS PIPE_CONTROL NOTIFY (CS_PIPE_CONTROL_Notify):** The field 'CS PIPE_CONTROL Notify' in register 'GT 0 Interrupts 2' does not have a description in the BXML
3	0h	**CS ERROR INTERRUPT (CS_Error_Interrupt):** The field 'CS Error Interrupt' in register 'GT 0 Interrupts 2' does not have a description in the BXML
2	0h	**SPARE 2 (Spare_2):** The field 'Spare 2' in register 'GT 0 Interrupts 2' does not have a description in the BXML
1	0h RO	Reserved.
0	0h	**CS MI USER INTERRUPT (CS_MI_User_Interrupt):** The field 'CS MI User Interrupt' in register 'GT 0 Interrupts 2' does not have a description in the BXML

32.16 GT 0 Interrupts 3 (GT_0_INTERRUPT_3)—Offset 4430Ch

This table indicates which events are mapped to each bit of the GT Interrupt 0 registers. Bits 15:0 are used for Render CS. Bits 31:16 are used for Blitter CS. The IER enabled Render Interrupt IIR (sticky) bits are ORed together to generate the Render Interrupts Pending bit in the Master Interrupt Control register. The IER enabled Blitter Interrupt IIR (sticky) bits are ORed together to generate the Blitter Interrupts Pending bit in the Master Interrupt Control register. 0x44300 = ISR 0x44304 = IMR 0x44308 = IIR 0x4430C = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h	**SPARE 31 (Spare_31):** The field 'Spare 31' in register 'GT 0 Interrupts 3' does not have a description in the BXML
30	0h	**SPARE 30 (Spare_30):** The field 'Spare 30' in register 'GT 0 Interrupts 3' does not have a description in the BXML
29	0h	**SPARE 29 (Spare_29):** The field 'Spare 29' in register 'GT 0 Interrupts 3' does not have a description in the BXML
28	0h	**SPARE 28 (Spare_28):** The field 'Spare 28' in register 'GT 0 Interrupts 3' does not have a description in the BXML
27	0h	**BCS WAIT ON SEMAPHORE (BCS_Wait_On_Semaphore):** The field 'BCS Wait On Semaphore' in register 'GT 0 Interrupts 3' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
26	0h	**SPARE 26 (Spare_26):** The field 'Spare 26' in register 'GT 0 Interrupts 3' does not have a description in the BXML
25	0h	**SPARE 25 (Spare_25):** The field 'Spare 25' in register 'GT 0 Interrupts 3' does not have a description in the BXML
24	0h	**BCS CONTEXT SWITCH INTERRUPT (BCS_Context_Switch_Interrupt):** The field 'BCS Context Switch Interrupt' in register 'GT 0 Interrupts 3' does not have a description in the BXML
23	0h	**SPARE 23 (Spare_23):** The field 'Spare 23' in register 'GT 0 Interrupts 3' does not have a description in the BXML
22	0h RO	Reserved.
21	0h	**SPARE 21 (Spare_21):** The field 'Spare 21' in register 'GT 0 Interrupts 3' does not have a description in the BXML
20	0h	**BCS MI FLUSH DW NOTIFY (BCS_MI_Flush_DW_Notify):** The field 'BCS MI Flush DW Notify' in register 'GT 0 Interrupts 3' does not have a description in the BXML
19	0h	**BCS ERROR INTERRUPT (BCS_Error_Interrupt):** The field 'BCS Error Interrupt' in register 'GT 0 Interrupts 3' does not have a description in the BXML
18	0h	**SPARE 18 (Spare_18):** The field 'Spare 18' in register 'GT 0 Interrupts 3' does not have a description in the BXML
17	0h	**SPARE 17 (Spare_17):** The field 'Spare 17' in register 'GT 0 Interrupts 3' does not have a description in the BXML
16	0h	**BCS MI USER INTERRUPT (BCS_MI_User_Interrupt):** The field 'BCS MI User Interrupt' in register 'GT 0 Interrupts 3' does not have a description in the BXML
15	0h	**SPARE 15 (Spare_15):** The field 'Spare 15' in register 'GT 0 Interrupts 3' does not have a description in the BXML
14	0h	**SPARE 14 (Spare_14):** The field 'Spare 14' in register 'GT 0 Interrupts 3' does not have a description in the BXML
13	0h	**SPARE 13 (Spare_13):** The field 'Spare 13' in register 'GT 0 Interrupts 3' does not have a description in the BXML
12	0h	**SPARE 12 (Spare_12):** The field 'Spare 12' in register 'GT 0 Interrupts 3' does not have a description in the BXML
11	0h	**CS WAIT ON SEMAPHORE (CS_Wait_On_Semaphore):** The field 'CS Wait On Semaphore' in register 'GT 0 Interrupts 3' does not have a description in the BXML
10	0h	**CS L3 COUNTER SAVE (CS_L3_Counter_Save):** The field 'CS L3 Counter Save' in register 'GT 0 Interrupts 3' does not have a description in the BXML
9	0h	**CS TR INVALID TILE DETECTION (CS_TR_Invalid_Tile_Detection):** The field 'CS TR Invalid Tile Detection' in register 'GT 0 Interrupts 3' does not have a description in the BXML
8	0h	**CS CONTEXT SWITCH INTERRUPT (CS_Context_Switch_Interrupt):** The field 'CS Context Switch Interrupt' in register 'GT 0 Interrupts 3' does not have a description in the BXML
7	0h	**PAGE FAULT INTERRUPT (Page_Fault_Interrupt):** This interrupt is for handling Legacy Page Fault interface for all Command Streamers [BCS, RCS, VCS, VECS]. When Fault Repair Mode is enabled, Interrupt mask register value is not looked at to generate interrupt due to page fault. Please refer to vol1c 'page fault support' section for more details. In Advanced (PRQ) Fault Interface is done through GUC interface.
6	0h	**CS WATCHDOG COUNTER EXPIRED (CS_Watchdog_Counter_Expired):** The field 'CS Watchdog Counter Expired' in register 'GT 0 Interrupts 3' does not have a description in the BXML
5	0h	**SPARE 5 (Spare_5):** The field 'Spare 5' in register 'GT 0 Interrupts 3' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
4	0h	**CS PIPE_CONTROL NOTIFY (CS_PIPE_CONTROL_Notify):** The field 'CS PIPE_CONTROL Notify' in register 'GT 0 Interrupts 3' does not have a description in the BXML
3	0h	**CS ERROR INTERRUPT (CS_Error_Interrupt):** The field 'CS Error Interrupt' in register 'GT 0 Interrupts 3' does not have a description in the BXML
2	0h	**SPARE 2 (Spare_2):** The field 'Spare 2' in register 'GT 0 Interrupts 3' does not have a description in the BXML
1	0h RO	Reserved.
0	0h	**CS MI USER INTERRUPT (CS_MI_User_Interrupt):** The field 'CS MI User Interrupt' in register 'GT 0 Interrupts 3' does not have a description in the BXML

32.17 GT 1 Interrupts 0 (GT_1_INTERRUPT_0)—Offset 44310h

This table indicates which events are mapped to each bit of the GT Interrupt 1 registers. Bits 15:0 are used for VCS1. Bits 31:16 are used for VCS2. The VCS1 Interrupt IIR (sticky) bits are ORed together to generate the VCS1 Interrupts Pending bit in the Master Interrupt Control register. The VCS2 Interrupt IIR (sticky) bits are ORed together to generate the VCS2 Interrupts Pending bit in the Master Interrupt Control register. 0x44310 = ISR 0x44314 = IMR 0x44318 = IIR 0x4431C = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h	**SPARE 31 (Spare_31):** The field 'Spare 31' in register 'GT 1 Interrupts 0' does not have a description in the BXML
30	0h	**SPARE 30 (Spare_30):** The field 'Spare 30' in register 'GT 1 Interrupts 0' does not have a description in the BXML
29	0h	**SPARE 29 (Spare_29):** The field 'Spare 29' in register 'GT 1 Interrupts 0' does not have a description in the BXML
28	0h	**SPARE 28 (Spare_28):** The field 'Spare 28' in register 'GT 1 Interrupts 0' does not have a description in the BXML
27	0h	**VCS2 WAIT ON SEMAPHORE (VCS2_Wait_On_Semaphore):** The field 'VCS2 Wait On Semaphore' in register 'GT 1 Interrupts 0' does not have a description in the BXML
26	0h	**SPARE 26 (Spare_26):** The field 'Spare 26' in register 'GT 1 Interrupts 0' does not have a description in the BXML
25	0h RO	Reserved.
24	0h	**VCS2 CONTEXT SWITCH INTERRUPT (VCS2_Context_Switch_Interrupt):** The field 'VCS2 Context Switch Interrupt' in register 'GT 1 Interrupts 0' does not have a description in the BXML
23	0h	**SPARE 23 (Spare_23):** The field 'Spare 23' in register 'GT 1 Interrupts 0' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
22	0h	**VCS2 WATCHDOG COUNTER EXPIRED (VCS2_Watchdog_Counter_Expired):** The field 'VCS2 Watchdog Counter Expired' in register 'GT 1 Interrupts 0' does not have a description in the BXML
21	0h RO	Reserved.
20	0h	**VCS2 MI FLUSH DW NOTIFY (VCS2_MI_Flush_DW_Notify):** The field 'VCS2 MI Flush DW Notify' in register 'GT 1 Interrupts 0' does not have a description in the BXML
19	0h	**VCS2 ERROR INTERRUPT (VCS2_Error_Interrupt):** The field 'VCS2 Error Interrupt' in register 'GT 1 Interrupts 0' does not have a description in the BXML
18	0h	**SPARE 18 (Spare_18):** The field 'Spare 18' in register 'GT 1 Interrupts 0' does not have a description in the BXML
17	0h	**SPARE 17 (Spare_17):** The field 'Spare 17' in register 'GT 1 Interrupts 0' does not have a description in the BXML
16	0h	**VCS2 MI USER INTERRUPT (VCS2_MI_User_Interrupt):** The field 'VCS2 MI User Interrupt' in register 'GT 1 Interrupts 0' does not have a description in the BXML
15	0h	**SPARE 15 (Spare_15):** The field 'Spare 15' in register 'GT 1 Interrupts 0' does not have a description in the BXML
14	0h	**SPARE 14 (Spare_14):** The field 'Spare 14' in register 'GT 1 Interrupts 0' does not have a description in the BXML
13	0h	**SPARE 13 (Spare_13):** The field 'Spare 13' in register 'GT 1 Interrupts 0' does not have a description in the BXML
12	0h	**SPARE 12 (Spare_12):** The field 'Spare 12' in register 'GT 1 Interrupts 0' does not have a description in the BXML
11	0h	**VCS1 WAIT ON SEMAPHORE (VCS1_Wait_On_Semaphore):** The field 'VCS1 Wait On Semaphore' in register 'GT 1 Interrupts 0' does not have a description in the BXML
10	0h	**SPARE 10 (Spare_10):** The field 'Spare 10' in register 'GT 1 Interrupts 0' does not have a description in the BXML
9	0h RO	Reserved.
8	0h	**VCS1 CONTEXT SWITCH INTERRUPT (VCS1_Context_Switch_Interrupt):** The field 'VCS1 Context Switch Interrupt' in register 'GT 1 Interrupts 0' does not have a description in the BXML
7	0h	**SPARE 7 (Spare_7):** The field 'Spare 7' in register 'GT 1 Interrupts 0' does not have a description in the BXML
6	0h	**VCS1 WATCHDOG COUNTER EXPIRED (VCS1_Watchdog_Counter_Expired):** The field 'VCS1 Watchdog Counter Expired' in register 'GT 1 Interrupts 0' does not have a description in the BXML
5	0h RO	Reserved.
4	0h	**VCS1 MI FLUSH DW NOTIFY (VCS1_MI_Flush_DW_Notify):** The field 'VCS1 MI Flush DW Notify' in register 'GT 1 Interrupts 0' does not have a description in the BXML
3	0h	**VCS1 ERROR INTERRUPT (VCS1_Error_Interrupt):** The field 'VCS1 Error Interrupt' in register 'GT 1 Interrupts 0' does not have a description in the BXML
2	0h	**SPARE 2 (Spare_2):** The field 'Spare 2' in register 'GT 1 Interrupts 0' does not have a description in the BXML
1	0h	**SPARE 1 (Spare_1):** The field 'Spare 1' in register 'GT 1 Interrupts 0' does not have a description in the BXML
0	0h	**VCS1 MI USER INTERRUPT (VCS1_MI_User_Interrupt):** The field 'VCS1 MI User Interrupt' in register 'GT 1 Interrupts 0' does not have a description in the BXML

32.18 GT 1 Interrupts 1 (GT_1_INTERRUPT_1)—Offset 44314h

This table indicates which events are mapped to each bit of the GT Interrupt 1 registers. Bits 15:0 are used for VCS1. Bits 31:16 are used for VCS2. The VCS1 Interrupt IIR (sticky) bits are ORed together to generate the VCS1 Interrupts Pending bit in the Master Interrupt Control register. The VCS2 Interrupt IIR (sticky) bits are ORed together to generate the VCS2 Interrupts Pending bit in the Master Interrupt Control register. 0x44310 = ISR 0x44314 = IMR 0x44318 = IIR 0x4431C = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h	**SPARE 31 (Spare_31):** The field 'Spare 31' in register 'GT 1 Interrupts 1' does not have a description in the BXML
30	0h	**SPARE 30 (Spare_30):** The field 'Spare 30' in register 'GT 1 Interrupts 1' does not have a description in the BXML
29	0h	**SPARE 29 (Spare_29):** The field 'Spare 29' in register 'GT 1 Interrupts 1' does not have a description in the BXML
28	0h	**SPARE 28 (Spare_28):** The field 'Spare 28' in register 'GT 1 Interrupts 1' does not have a description in the BXML
27	0h	**VCS2 WAIT ON SEMAPHORE (VCS2_Wait_On_Semaphore):** The field 'VCS2 Wait On Semaphore' in register 'GT 1 Interrupts 1' does not have a description in the BXML
26	0h	**SPARE 26 (Spare_26):** The field 'Spare 26' in register 'GT 1 Interrupts 1' does not have a description in the BXML
25	0h RO	Reserved.
24	0h	**VCS2 CONTEXT SWITCH INTERRUPT (VCS2_Context_Switch_Interrupt):** The field 'VCS2 Context Switch Interrupt' in register 'GT 1 Interrupts 1' does not have a description in the BXML
23	0h	**SPARE 23 (Spare_23):** The field 'Spare 23' in register 'GT 1 Interrupts 1' does not have a description in the BXML
22	0h	**VCS2 WATCHDOG COUNTER EXPIRED (VCS2_Watchdog_Counter_Expired):** The field 'VCS2 Watchdog Counter Expired' in register 'GT 1 Interrupts 1' does not have a description in the BXML
21	0h RO	Reserved.
20	0h	**VCS2 MI FLUSH DW NOTIFY (VCS2_MI_Flush_DW_Notify):** The field 'VCS2 MI Flush DW Notify' in register 'GT 1 Interrupts 1' does not have a description in the BXML
19	0h	**VCS2 ERROR INTERRUPT (VCS2_Error_Interrupt):** The field 'VCS2 Error Interrupt' in register 'GT 1 Interrupts 1' does not have a description in the BXML
18	0h	**SPARE 18 (Spare_18):** The field 'Spare 18' in register 'GT 1 Interrupts 1' does not have a description in the BXML
17	0h	**SPARE 17 (Spare_17):** The field 'Spare 17' in register 'GT 1 Interrupts 1' does not have a description in the BXML
16	0h	**VCS2 MI USER INTERRUPT (VCS2_MI_User_Interrupt):** The field 'VCS2 MI User Interrupt' in register 'GT 1 Interrupts 1' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
15	0h	**SPARE 15 (Spare_15):** The field 'Spare 15' in register 'GT 1 Interrupts 1' does not have a description in the BXML
14	0h	**SPARE 14 (Spare_14):** The field 'Spare 14' in register 'GT 1 Interrupts 1' does not have a description in the BXML
13	0h	**SPARE 13 (Spare_13):** The field 'Spare 13' in register 'GT 1 Interrupts 1' does not have a description in the BXML
12	0h	**SPARE 12 (Spare_12):** The field 'Spare 12' in register 'GT 1 Interrupts 1' does not have a description in the BXML
11	0h	**VCS1 WAIT ON SEMAPHORE (VCS1_Wait_On_Semaphore):** The field 'VCS1 Wait On Semaphore' in register 'GT 1 Interrupts 1' does not have a description in the BXML
10	0h	**SPARE 10 (Spare_10):** The field 'Spare 10' in register 'GT 1 Interrupts 1' does not have a description in the BXML
9	0h RO	Reserved.
8	0h	**VCS1 CONTEXT SWITCH INTERRUPT (VCS1_Context_Switch_Interrupt):** The field 'VCS1 Context Switch Interrupt' in register 'GT 1 Interrupts 1' does not have a description in the BXML
7	0h	**SPARE 7 (Spare_7):** The field 'Spare 7' in register 'GT 1 Interrupts 1' does not have a description in the BXML
6	0h	**VCS1 WATCHDOG COUNTER EXPIRED (VCS1_Watchdog_Counter_Expired):** The field 'VCS1 Watchdog Counter Expired' in register 'GT 1 Interrupts 1' does not have a description in the BXML
5	0h RO	Reserved.
4	0h	**VCS1 MI FLUSH DW NOTIFY (VCS1_MI_Flush_DW_Notify):** The field 'VCS1 MI Flush DW Notify' in register 'GT 1 Interrupts 1' does not have a description in the BXML
3	0h	**VCS1 ERROR INTERRUPT (VCS1_Error_Interrupt):** The field 'VCS1 Error Interrupt' in register 'GT 1 Interrupts 1' does not have a description in the BXML
2	0h	**SPARE 2 (Spare_2):** The field 'Spare 2' in register 'GT 1 Interrupts 1' does not have a description in the BXML
1	0h	**SPARE 1 (Spare_1):** The field 'Spare 1' in register 'GT 1 Interrupts 1' does not have a description in the BXML
0	0h	**VCS1 MI USER INTERRUPT (VCS1_MI_User_Interrupt):** The field 'VCS1 MI User Interrupt' in register 'GT 1 Interrupts 1' does not have a description in the BXML

32.19 GT 1 Interrupts 2 (GT_1_INTERRUPT_2)—Offset 44318h

This table indicates which events are mapped to each bit of the GT Interrupt 1 registers. Bits 15:0 are used for VCS1. Bits 31:16 are used for VCS2. The VCS1 Interrupt IIR (sticky) bits are ORed together to generate the VCS1 Interrupts Pending bit in the Master Interrupt Control register. The VCS2 Interrupt IIR (sticky) bits are ORed together to generate the VCS2 Interrupts Pending bit in the Master Interrupt Control register. 0x44310 = ISR 0x44314 = IMR 0x44318 = IIR 0x4431C = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h	**SPARE 31 (Spare_31):** The field 'Spare 31' in register 'GT 1 Interrupts 2' does not have a description in the BXML
30	0h	**SPARE 30 (Spare_30):** The field 'Spare 30' in register 'GT 1 Interrupts 2' does not have a description in the BXML
29	0h	**SPARE 29 (Spare_29):** The field 'Spare 29' in register 'GT 1 Interrupts 2' does not have a description in the BXML
28	0h	**SPARE 28 (Spare_28):** The field 'Spare 28' in register 'GT 1 Interrupts 2' does not have a description in the BXML
27	0h	**VCS2 WAIT ON SEMAPHORE (VCS2_Wait_On_Semaphore):** The field 'VCS2 Wait On Semaphore' in register 'GT 1 Interrupts 2' does not have a description in the BXML
26	0h	**SPARE 26 (Spare_26):** The field 'Spare 26' in register 'GT 1 Interrupts 2' does not have a description in the BXML
25	0h RO	Reserved.
24	0h	**VCS2 CONTEXT SWITCH INTERRUPT (VCS2_Context_Switch_Interrupt):** The field 'VCS2 Context Switch Interrupt' in register 'GT 1 Interrupts 2' does not have a description in the BXML
23	0h	**SPARE 23 (Spare_23):** The field 'Spare 23' in register 'GT 1 Interrupts 2' does not have a description in the BXML
22	0h	**VCS2 WATCHDOG COUNTER EXPIRED (VCS2_Watchdog_Counter_Expired):** The field 'VCS2 Watchdog Counter Expired' in register 'GT 1 Interrupts 2' does not have a description in the BXML
21	0h RO	Reserved.
20	0h	**VCS2 MI FLUSH DW NOTIFY (VCS2_MI_Flush_DW_Notify):** The field 'VCS2 MI Flush DW Notify' in register 'GT 1 Interrupts 2' does not have a description in the BXML
19	0h	**VCS2 ERROR INTERRUPT (VCS2_Error_Interrupt):** The field 'VCS2 Error Interrupt' in register 'GT 1 Interrupts 2' does not have a description in the BXML
18	0h	**SPARE 18 (Spare_18):** The field 'Spare 18' in register 'GT 1 Interrupts 2' does not have a description in the BXML
17	0h	**SPARE 17 (Spare_17):** The field 'Spare 17' in register 'GT 1 Interrupts 2' does not have a description in the BXML
16	0h	**VCS2 MI USER INTERRUPT (VCS2_MI_User_Interrupt):** The field 'VCS2 MI User Interrupt' in register 'GT 1 Interrupts 2' does not have a description in the BXML
15	0h	**SPARE 15 (Spare_15):** The field 'Spare 15' in register 'GT 1 Interrupts 2' does not have a description in the BXML
14	0h	**SPARE 14 (Spare_14):** The field 'Spare 14' in register 'GT 1 Interrupts 2' does not have a description in the BXML
13	0h	**SPARE 13 (Spare_13):** The field 'Spare 13' in register 'GT 1 Interrupts 2' does not have a description in the BXML
12	0h	**SPARE 12 (Spare_12):** The field 'Spare 12' in register 'GT 1 Interrupts 2' does not have a description in the BXML
11	0h	**VCS1 WAIT ON SEMAPHORE (VCS1_Wait_On_Semaphore):** The field 'VCS1 Wait On Semaphore' in register 'GT 1 Interrupts 2' does not have a description in the BXML
10	0h	**SPARE 10 (Spare_10):** The field 'Spare 10' in register 'GT 1 Interrupts 2' does not have a description in the BXML
9	0h RO	Reserved.

Bit Range	Default & Access	Field Name (ID): Description
8	0h	**VCS1 CONTEXT SWITCH INTERRUPT (VCS1_Context_Switch_Interrupt):** The field 'VCS1 Context Switch Interrupt' in register 'GT 1 Interrupts 2' does not have a description in the BXML
7	0h	**SPARE 7 (Spare_7):** The field 'Spare 7' in register 'GT 1 Interrupts 2' does not have a description in the BXML
6	0h	**VCS1 WATCHDOG COUNTER EXPIRED (VCS1_Watchdog_Counter_Expired):** The field 'VCS1 Watchdog Counter Expired' in register 'GT 1 Interrupts 2' does not have a description in the BXML
5	0h RO	Reserved.
4	0h	**VCS1 MI FLUSH DW NOTIFY (VCS1_MI_Flush_DW_Notify):** The field 'VCS1 MI Flush DW Notify' in register 'GT 1 Interrupts 2' does not have a description in the BXML
3	0h	**VCS1 ERROR INTERRUPT (VCS1_Error_Interrupt):** The field 'VCS1 Error Interrupt' in register 'GT 1 Interrupts 2' does not have a description in the BXML
2	0h	**SPARE 2 (Spare_2):** The field 'Spare 2' in register 'GT 1 Interrupts 2' does not have a description in the BXML
1	0h	**SPARE 1 (Spare_1):** The field 'Spare 1' in register 'GT 1 Interrupts 2' does not have a description in the BXML
0	0h	**VCS1 MI USER INTERRUPT (VCS1_MI_User_Interrupt):** The field 'VCS1 MI User Interrupt' in register 'GT 1 Interrupts 2' does not have a description in the BXML

32.20 GT 1 Interrupts 3 (GT_1_INTERRUPT_3)—Offset 4431Ch

This table indicates which events are mapped to each bit of the GT Interrupt 1 registers. Bits 15:0 are used for VCS1. Bits 31:16 are used for VCS2. The VCS1 Interrupt IIR (sticky) bits are ORed together to generate the VCS1 Interrupts Pending bit in the Master Interrupt Control register. The VCS2 Interrupt IIR (sticky) bits are ORed together to generate the VCS2 Interrupts Pending bit in the Master Interrupt Control register. 0x44310 = ISR 0x44314 = IMR 0x44318 = IIR 0x4431C = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h	**SPARE 31 (Spare_31):** The field 'Spare 31' in register 'GT 1 Interrupts 3' does not have a description in the BXML
30	0h	**SPARE 30 (Spare_30):** The field 'Spare 30' in register 'GT 1 Interrupts 3' does not have a description in the BXML
29	0h	**SPARE 29 (Spare_29):** The field 'Spare 29' in register 'GT 1 Interrupts 3' does not have a description in the BXML
28	0h	**SPARE 28 (Spare_28):** The field 'Spare 28' in register 'GT 1 Interrupts 3' does not have a description in the BXML
27	0h	**VCS2 WAIT ON SEMAPHORE (VCS2_Wait_On_Semaphore):** The field 'VCS2 Wait On Semaphore' in register 'GT 1 Interrupts 3' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
26	0h	**SPARE 26 (Spare_26):** The field 'Spare 26' in register 'GT 1 Interrupts 3' does not have a description in the BXML
25	0h RO	Reserved.
24	0h	**VCS2 CONTEXT SWITCH INTERRUPT (VCS2_Context_Switch_Interrupt):** The field 'VCS2 Context Switch Interrupt' in register 'GT 1 Interrupts 3' does not have a description in the BXML
23	0h	**SPARE 23 (Spare_23):** The field 'Spare 23' in register 'GT 1 Interrupts 3' does not have a description in the BXML
22	0h	**VCS2 WATCHDOG COUNTER EXPIRED (VCS2_Watchdog_Counter_Expired):** The field 'VCS2 Watchdog Counter Expired' in register 'GT 1 Interrupts 3' does not have a description in the BXML
21	0h RO	Reserved.
20	0h	**VCS2 MI FLUSH DW NOTIFY (VCS2_MI_Flush_DW_Notify):** The field 'VCS2 MI Flush DW Notify' in register 'GT 1 Interrupts 3' does not have a description in the BXML
19	0h	**VCS2 ERROR INTERRUPT (VCS2_Error_Interrupt):** The field 'VCS2 Error Interrupt' in register 'GT 1 Interrupts 3' does not have a description in the BXML
18	0h	**SPARE 18 (Spare_18):** The field 'Spare 18' in register 'GT 1 Interrupts 3' does not have a description in the BXML
17	0h	**SPARE 17 (Spare_17):** The field 'Spare 17' in register 'GT 1 Interrupts 3' does not have a description in the BXML
16	0h	**VCS2 MI USER INTERRUPT (VCS2_MI_User_Interrupt):** The field 'VCS2 MI User Interrupt' in register 'GT 1 Interrupts 3' does not have a description in the BXML
15	0h	**SPARE 15 (Spare_15):** The field 'Spare 15' in register 'GT 1 Interrupts 3' does not have a description in the BXML
14	0h	**SPARE 14 (Spare_14):** The field 'Spare 14' in register 'GT 1 Interrupts 3' does not have a description in the BXML
13	0h	**SPARE 13 (Spare_13):** The field 'Spare 13' in register 'GT 1 Interrupts 3' does not have a description in the BXML
12	0h	**SPARE 12 (Spare_12):** The field 'Spare 12' in register 'GT 1 Interrupts 3' does not have a description in the BXML
11	0h	**VCS1 WAIT ON SEMAPHORE (VCS1_Wait_On_Semaphore):** The field 'VCS1 Wait On Semaphore' in register 'GT 1 Interrupts 3' does not have a description in the BXML
10	0h	**SPARE 10 (Spare_10):** The field 'Spare 10' in register 'GT 1 Interrupts 3' does not have a description in the BXML
9	0h RO	Reserved.
8	0h	**VCS1 CONTEXT SWITCH INTERRUPT (VCS1_Context_Switch_Interrupt):** The field 'VCS1 Context Switch Interrupt' in register 'GT 1 Interrupts 3' does not have a description in the BXML
7	0h	**SPARE 7 (Spare_7):** The field 'Spare 7' in register 'GT 1 Interrupts 3' does not have a description in the BXML
6	0h	**VCS1 WATCHDOG COUNTER EXPIRED (VCS1_Watchdog_Counter_Expired):** The field 'VCS1 Watchdog Counter Expired' in register 'GT 1 Interrupts 3' does not have a description in the BXML
5	0h RO	Reserved.
4	0h	**VCS1 MI FLUSH DW NOTIFY (VCS1_MI_Flush_DW_Notify):** The field 'VCS1 MI Flush DW Notify' in register 'GT 1 Interrupts 3' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
3	0h	**VCS1 ERROR INTERRUPT (VCS1_Error_Interrupt):** The field 'VCS1 Error Interrupt' in register 'GT 1 Interrupts 3' does not have a description in the BXML
2	0h	**SPARE 2 (Spare_2):** The field 'Spare 2' in register 'GT 1 Interrupts 3' does not have a description in the BXML
1	0h	**SPARE 1 (Spare_1):** The field 'Spare 1' in register 'GT 1 Interrupts 3' does not have a description in the BXML
0	0h	**VCS1 MI USER INTERRUPT (VCS1_MI_User_Interrupt):** The field 'VCS1 MI User Interrupt' in register 'GT 1 Interrupts 3' does not have a description in the BXML

32.21 GT 2 Interrupts 0 (GT_2_INTERRUPT_0)—Offset 44320h

This table indicates which events are mapped to each bit of the GT Interrupt 2 registers. Bits 15:0 are used for GTPM. The IER enabled GTPM Interrupt IIR (sticky) bits are ORed together to generate the GTPM Interrupts Pending bit in the Master Interrupt Control register. 0x44320 = ISR 0x44324 = IMR 0x44328 = IIR 0x4432C = IER Bits 31:16 are used for GuC. The IER enabled GuC Interrupt IIR (sticky) bits are ORed together to generate the GuC Interrupts Pending bit in the Master Interrupt Control register.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	Reserved.
15	0h	**SPARE 15 (Spare_15):** The field 'Spare 15' in register 'GT 2 Interrupts 0' does not have a description in the BXML
14	0h	**SPARE 14 (Spare_14):** The field 'Spare 14' in register 'GT 2 Interrupts 0' does not have a description in the BXML
13	0h	**UNSLICE FREQUENCY CONTROL UP INTERRUPT (Unslice_Frequency_Control_Up_Interrupt):** The field 'Unslice Frequency Control Up Interrupt' in register 'GT 2 Interrupts 0' does not have a description in the BXML
12	0h	**UNSLICE FREQUENCY CONTROL DOWN INTERRUPT (Unslice_Frequency_Control_Down_Interrupt):** The field 'Unslice Frequency Control Down Interrupt' in register 'GT 2 Interrupts 0' does not have a description in the BXML
11	0h	**NFADFL FREQUENCY UP INTERRUPT (NFADFL_Frequency_Up_Interrupt):** The field 'NFADFL Frequency Up Interrupt' in register 'GT 2 Interrupts 0' does not have a description in the BXML
10	0h	**NFADFL FREQUENCY DOWN INTERRUPT (NFADFL_Frequency_Down_Interrupt):** The field 'NFADFL Frequency Down Interrupt' in register 'GT 2 Interrupts 0' does not have a description in the BXML
9	0h RO	Reserved.
8	0h	**GTPM ENGINES IDLE INTERRUPT (GTPM_Engines_Idle_Interrupt):** The field 'GTPM Engines Idle Interrupt' in register 'GT 2 Interrupts 0' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
7	0h	**GTPM UNCORE TO CORE TRAP INTERRUPT (GTPM_Uncore_to_Core_Trap_Interrupt):** The field 'GTPM Uncore to Core Trap Interrupt' in register 'GT 2 Interrupts 0' does not have a description in the BXML
6	0h	**GTPM RENDER FREQUENCY DOWNWARDS TIMEOUT DURING RC6 INTERRUPT (GTPM_Render_Frequency_Downwards_Timeout_During_RC6_Interrupt):** The field 'GTPM Render Frequency Downwards Timeout During RC6 Interrupt' in register 'GT 2 Interrupts 0' does not have a description in the BXML
5	0h	**GTPM RENDER P-STATE UP THRESHOLD INTERRUPT (GTPM_Render_PState_Up_Threshold_Interrupt):** The field 'GTPM Render P-State Up Threshold Interrupt' in register 'GT 2 Interrupts 0' does not have a description in the BXML
4	0h	**GTPM RENDER P-STATE DOWN THRESHOLD INTERRUPT (GTPM_Render_PState_Down_Threshold_Interrupt):** The field 'GTPM Render P-State Down Threshold Interrupt' in register 'GT 2 Interrupts 0' does not have a description in the BXML
3	0h	**SPARE 3 (Spare_3):** The field 'Spare 3' in register 'GT 2 Interrupts 0' does not have a description in the BXML
2	0h	**GTPM RENDER GEYSERVILLE UP EVALUATION INTERVAL INTERRUPT (GTPM_Render_Geyserville_Up_Evaluation_Interval_Interrupt):** The field 'GTPM Render Geyserville Up Evaluation Interval Interrupt' in register 'GT 2 Interrupts 0' does not have a description in the BXML
1	0h	**GTPM RENDER GEYSERVILLE DOWN EVALUATION INTERVAL INTERRUPT (GTPM_Render_Geyserville_Down_Evaluation_Interval_Interrupt):** The field 'GTPM Render Geyserville Down Evaluation Interval Interrupt' in register 'GT 2 Interrupts 0' does not have a description in the BXML
0	0h	**SPARE 0 (Spare_0):** The field 'Spare 0' in register 'GT 2 Interrupts 0' does not have a description in the BXML

32.22 GT 2 Interrupts 1 (GT_2_INTERRUPT_1)—Offset 44324h

This table indicates which events are mapped to each bit of the GT Interrupt 2 registers. Bits 15:0 are used for GTPM. The IER enabled GTPM Interrupt IIR (sticky) bits are ORed together to generate the GTPM Interrupts Pending bit in the Master Interrupt Control register. 0x44320 = ISR 0x44324 = IMR 0x44328 = IIR 0x4432C = IER Bits 31:16 are used for GuC. The IER enabled GuC Interrupt IIR (sticky) bits are ORed together to generate the GuC Interrupts Pending bit in the Master Interrupt Control register.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	Reserved.
15	0h	**SPARE 15 (Spare_15):** The field 'Spare 15' in register 'GT 2 Interrupts 1' does not have a description in the BXML
14	0h	**SPARE 14 (Spare_14):** The field 'Spare 14' in register 'GT 2 Interrupts 1' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
13	0h	**UNSLICE FREQUENCY CONTROL UP INTERRUPT (Unslice_Frequency_Control_Up_Interrupt):** The field 'Unslice Frequency Control Up Interrupt' in register 'GT 2 Interrupts 1' does not have a description in the BXML
12	0h	**UNSLICE FREQUENCY CONTROL DOWN INTERRUPT (Unslice_Frequency_Control_Down_Interrupt):** The field 'Unslice Frequency Control Down Interrupt' in register 'GT 2 Interrupts 1' does not have a description in the BXML
11	0h	**NFADFL FREQUENCY UP INTERRUPT (NFADFL_Frequency_Up_Interrupt):** The field 'NFADFL Frequency Up Interrupt' in register 'GT 2 Interrupts 1' does not have a description in the BXML
10	0h	**NFADFL FREQUENCY DOWN INTERRUPT (NFADFL_Frequency_Down_Interrupt):** The field 'NFADFL Frequency Down Interrupt' in register 'GT 2 Interrupts 1' does not have a description in the BXML
9	0h RO	Reserved.
8	0h	**GTPM ENGINES IDLE INTERRUPT (GTPM_Engines_Idle_Interrupt):** The field 'GTPM Engines Idle Interrupt' in register 'GT 2 Interrupts 1' does not have a description in the BXML
7	0h	**GTPM UNCORE TO CORE TRAP INTERRUPT (GTPM_Uncore_to_Core_Trap_Interrupt):** The field 'GTPM Uncore to Core Trap Interrupt' in register 'GT 2 Interrupts 1' does not have a description in the BXML
6	0h	**GTPM RENDER FREQUENCY DOWNWARDS TIMEOUT DURING RC6 INTERRUPT (GTPM_Render_Frequency_Downwards_Timeout_During_RC6_Interrupt):** The field 'GTPM Render Frequency Downwards Timeout During RC6 Interrupt' in register 'GT 2 Interrupts 1' does not have a description in the BXML
5	0h	**GTPM RENDER P-STATE UP THRESHOLD INTERRUPT (GTPM_Render_PState_Up_Threshold_Interrupt):** The field 'GTPM Render P-State Up Threshold Interrupt' in register 'GT 2 Interrupts 1' does not have a description in the BXML
4	0h	**GTPM RENDER P-STATE DOWN THRESHOLD INTERRUPT (GTPM_Render_PState_Down_Threshold_Interrupt):** The field 'GTPM Render P-State Down Threshold Interrupt' in register 'GT 2 Interrupts 1' does not have a description in the BXML
3	0h	**SPARE 3 (Spare_3):** The field 'Spare 3' in register 'GT 2 Interrupts 1' does not have a description in the BXML
2	0h	**GTPM RENDER GEYSERVILLE UP EVALUATION INTERVAL INTERRUPT (GTPM_Render_Geyserville_Up_Evaluation_Interval_Interrupt):** The field 'GTPM Render Geyserville Up Evaluation Interval Interrupt' in register 'GT 2 Interrupts 1' does not have a description in the BXML
1	0h	**GTPM RENDER GEYSERVILLE DOWN EVALUATION INTERVAL INTERRUPT (GTPM_Render_Geyserville_Down_Evaluation_Interval_Interrupt):** The field 'GTPM Render Geyserville Down Evaluation Interval Interrupt' in register 'GT 2 Interrupts 1' does not have a description in the BXML
0	0h	**SPARE 0 (Spare_0):** The field 'Spare 0' in register 'GT 2 Interrupts 1' does not have a description in the BXML

32.23 GT 2 Interrupts 2 (GT_2_INTERRUPT_2)—Offset 44328h

This table indicates which events are mapped to each bit of the GT Interrupt 2 registers. Bits 15:0 are used for GTPM. The IER enabled GTPM Interrupt IIR (sticky) bits are ORed together to generate the GTPM Interrupts Pending bit in the Master Interrupt Control register. 0x44320 = ISR 0x44324 = IMR 0x44328 = IIR 0x4432C = IER Bits 31:16 are used for GuC. The IER enabled GuC Interrupt IIR (sticky) bits are ORed together to generate the GuC Interrupts Pending bit in the Master Interrupt Control register.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	Reserved.
15	0h	**SPARE 15 (Spare_15):** The field 'Spare 15' in register 'GT 2 Interrupts 2' does not have a description in the BXML
14	0h	**SPARE 14 (Spare_14):** The field 'Spare 14' in register 'GT 2 Interrupts 2' does not have a description in the BXML
13	0h	**UNSLICE FREQUENCY CONTROL UP INTERRUPT (Unslice_Frequency_Control_Up_Interrupt):** The field 'Unslice Frequency Control Up Interrupt' in register 'GT 2 Interrupts 2' does not have a description in the BXML
12	0h	**UNSLICE FREQUENCY CONTROL DOWN INTERRUPT (Unslice_Frequency_Control_Down_Interrupt):** The field 'Unslice Frequency Control Down Interrupt' in register 'GT 2 Interrupts 2' does not have a description in the BXML
11	0h	**NFADFL FREQUENCY UP INTERRUPT (NFADFL_Frequency_Up_Interrupt):** The field 'NFADFL Frequency Up Interrupt' in register 'GT 2 Interrupts 2' does not have a description in the BXML
10	0h	**NFADFL FREQUENCY DOWN INTERRUPT (NFADFL_Frequency_Down_Interrupt):** The field 'NFADFL Frequency Down Interrupt' in register 'GT 2 Interrupts 2' does not have a description in the BXML
9	0h RO	Reserved.
8	0h	**GTPM ENGINES IDLE INTERRUPT (GTPM_Engines_Idle_Interrupt):** The field 'GTPM Engines Idle Interrupt' in register 'GT 2 Interrupts 2' does not have a description in the BXML
7	0h	**GTPM UNCORE TO CORE TRAP INTERRUPT (GTPM_Uncore_to_Core_Trap_Interrupt):** The field 'GTPM Uncore to Core Trap Interrupt' in register 'GT 2 Interrupts 2' does not have a description in the BXML
6	0h	**GTPM RENDER FREQUENCY DOWNWARDS TIMEOUT DURING RC6 INTERRUPT (GTPM_Render_Frequency_Downwards_Timeout_During_RC6_Interrupt):** The field 'GTPM Render Frequency Downwards Timeout During RC6 Interrupt' in register 'GT 2 Interrupts 2' does not have a description in the BXML
5	0h	**GTPM RENDER P-STATE UP THRESHOLD INTERRUPT (GTPM_Render_PState_Up_Threshold_Interrupt):** The field 'GTPM Render P-State Up Threshold Interrupt' in register 'GT 2 Interrupts 2' does not have a description in the BXML
4	0h	**GTPM RENDER P-STATE DOWN THRESHOLD INTERRUPT (GTPM_Render_PState_Down_Threshold_Interrupt):** The field 'GTPM Render P-State Down Threshold Interrupt' in register 'GT 2 Interrupts 2' does not have a description in the BXML
3	0h	**SPARE 3 (Spare_3):** The field 'Spare 3' in register 'GT 2 Interrupts 2' does not have a description in the BXML
2	0h	**GTPM RENDER GEYSERVILLE UP EVALUATION INTERVAL INTERRUPT (GTPM_Render_Geyserville_Up_Evaluation_Interval_Interrupt):** The field 'GTPM Render Geyserville Up Evaluation Interval Interrupt' in register 'GT 2 Interrupts 2' does not have a description in the BXML
1	0h	**GTPM RENDER GEYSERVILLE DOWN EVALUATION INTERVAL INTERRUPT (GTPM_Render_Geyserville_Down_Evaluation_Interval_Interrupt):** The field 'GTPM Render Geyserville Down Evaluation Interval Interrupt' in register 'GT 2 Interrupts 2' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
0	0h	**SPARE 0 (Spare_0):** The field 'Spare 0' in register 'GT 2 Interrupts 2' does not have a description in the BXML

32.24 GT 2 Interrupts 3 (GT_2_INTERRUPT_3)—Offset 4432Ch

This table indicates which events are mapped to each bit of the GT Interrupt 2 registers. Bits 15:0 are used for GTPM. The IER enabled GTPM Interrupt IIR (sticky) bits are ORed together to generate the GTPM Interrupts Pending bit in the Master Interrupt Control register. 0x44320 = ISR 0x44324 = IMR 0x44328 = IIR 0x4432C = IER Bits 31:16 are used for GuC. The IER enabled GuC Interrupt IIR (sticky) bits are ORed together to generate the GuC Interrupts Pending bit in the Master Interrupt Control register.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	Reserved.
15	0h	**SPARE 15 (Spare_15):** The field 'Spare 15' in register 'GT 2 Interrupts 3' does not have a description in the BXML
14	0h	**SPARE 14 (Spare_14):** The field 'Spare 14' in register 'GT 2 Interrupts 3' does not have a description in the BXML
13	0h	**UNSLICE FREQUENCY CONTROL UP INTERRUPT (Unslice_Frequency_Control_Up_Interrupt):** The field 'Unslice Frequency Control Up Interrupt' in register 'GT 2 Interrupts 3' does not have a description in the BXML
12	0h	**UNSLICE FREQUENCY CONTROL DOWN INTERRUPT (Unslice_Frequency_Control_Down_Interrupt):** The field 'Unslice Frequency Control Down Interrupt' in register 'GT 2 Interrupts 3' does not have a description in the BXML
11	0h	**NFADFL FREQUENCY UP INTERRUPT (NFADFL_Frequency_Up_Interrupt):** The field 'NFADFL Frequency Up Interrupt' in register 'GT 2 Interrupts 3' does not have a description in the BXML
10	0h	**NFADFL FREQUENCY DOWN INTERRUPT (NFADFL_Frequency_Down_Interrupt):** The field 'NFADFL Frequency Down Interrupt' in register 'GT 2 Interrupts 3' does not have a description in the BXML
9	0h RO	Reserved.
8	0h	**GTPM ENGINES IDLE INTERRUPT (GTPM_Engines_Idle_Interrupt):** The field 'GTPM Engines Idle Interrupt' in register 'GT 2 Interrupts 3' does not have a description in the BXML
7	0h	**GTPM UNCORE TO CORE TRAP INTERRUPT (GTPM_Uncore_to_Core_Trap_Interrupt):** The field 'GTPM Uncore to Core Trap Interrupt' in register 'GT 2 Interrupts 3' does not have a description in the BXML
6	0h	**GTPM RENDER FREQUENCY DOWNWARDS TIMEOUT DURING RC6 INTERRUPT (GTPM_Render_Frequency_Downwards_Timeout_During_RC6_Interrupt):** The field 'GTPM Render Frequency Downwards Timeout During RC6 Interrupt' in register 'GT 2 Interrupts 3' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
5	0h	**GTPM RENDER P-STATE UP THRESHOLD INTERRUPT (GTPM_Render_PState_Up_Threshold_Interrupt):** The field 'GTPM Render P-State Up Threshold Interrupt' in register 'GT 2 Interrupts 3' does not have a description in the BXML
4	0h	**GTPM RENDER P-STATE DOWN THRESHOLD INTERRUPT (GTPM_Render_PState_Down_Threshold_Interrupt):** The field 'GTPM Render P-State Down Threshold Interrupt' in register 'GT 2 Interrupts 3' does not have a description in the BXML
3	0h	**SPARE 3 (Spare_3):** The field 'Spare 3' in register 'GT 2 Interrupts 3' does not have a description in the BXML
2	0h	**GTPM RENDER GEYSERVILLE UP EVALUATION INTERVAL INTERRUPT (GTPM_Render_Geyserville_Up_Evaluation_Interval_Interrupt):** The field 'GTPM Render Geyserville Up Evaluation Interval Interrupt' in register 'GT 2 Interrupts 3' does not have a description in the BXML
1	0h	**GTPM RENDER GEYSERVILLE DOWN EVALUATION INTERVAL INTERRUPT (GTPM_Render_Geyserville_Down_Evaluation_Interval_Interrupt):** The field 'GTPM Render Geyserville Down Evaluation Interval Interrupt' in register 'GT 2 Interrupts 3' does not have a description in the BXML
0	0h	**SPARE 0 (Spare_0):** The field 'Spare 0' in register 'GT 2 Interrupts 3' does not have a description in the BXML

32.25 GT 3 Interrupts 0 (GT_3_INTERRUPT_0)—Offset 44330h

This table indicates which events are mapped to each bit of the GT Interrupt 3 registers. Bits 15:0 are used for VEBox. Bits 27:16 are used for WDBox. Bits 31:28 are used for OACS. The VEBox Interrupt IIR (sticky) bits are ORed together to generate the VEBox Interrupts Pending bit in the Master Interrupt Control register. The WDBox and OACS Interrupt IIR (sticky) bits are ORed together to generate the WDBox Interrupts Pending bit in the Master Interrupt Control register. 0x44330 = ISR 0x44334 = IMR 0x44338 = IIR 0x4433C = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h	**SPARE 31 (Spare_31):** The field 'Spare 31' in register 'GT 3 Interrupts 0' does not have a description in the BXML
30	0h	**SPARE 30 (Spare_30):** The field 'Spare 30' in register 'GT 3 Interrupts 0' does not have a description in the BXML
29	0h	**SPARE 29 (Spare_29):** The field 'Spare 29' in register 'GT 3 Interrupts 0' does not have a description in the BXML
28	0h	**PERFORMANCE MONITORING BUFFER HALF-FULL INTERRUPT (Performance_Monitoring_Buffer_HalfFull_Interrupt):** For internal trigger (timer event based) reporting, this interrupt is generated if the report buffer crosses the half full limit. For internal trigger (NOA event based) reporting, this interrupt is generated if the report buffer crosses the half full limit.
27	0h	**SPARE 27 (Spare_27):** The field 'Spare 27' in register 'GT 3 Interrupts 0' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
26	0h	**SPARE 26 (Spare_26):** The field 'Spare 26' in register 'GT 3 Interrupts 0' does not have a description in the BXML
25	0h	**SPARE 25 (Spare_25):** The field 'Spare 25' in register 'GT 3 Interrupts 0' does not have a description in the BXML
24	0h	**SPARE 24 (Spare_24):** The field 'Spare 24' in register 'GT 3 Interrupts 0' does not have a description in the BXML
23	0h	**SPARE 23 (Spare_23):** The field 'Spare 23' in register 'GT 3 Interrupts 0' does not have a description in the BXML
22	0h	**SPARE 22 (Spare_22):** The field 'Spare 22' in register 'GT 3 Interrupts 0' does not have a description in the BXML
21	0h	**SPARE 21 (Spare_21):** The field 'Spare 21' in register 'GT 3 Interrupts 0' does not have a description in the BXML
20	0h	**SPARE 20 (Spare_20):** The field 'Spare 20' in register 'GT 3 Interrupts 0' does not have a description in the BXML
19	0h	**SPARE 19 (Spare_19):** The field 'Spare 19' in register 'GT 3 Interrupts 0' does not have a description in the BXML
18	0h	**SPARE 18 (Spare_18):** The field 'Spare 18' in register 'GT 3 Interrupts 0' does not have a description in the BXML
17	0h	**WDBOX 1 STATUS INTERRUPT (WDBox_1_Status_Interrupt):** The field 'WDBox 1 Status Interrupt' in register 'GT 3 Interrupts 0' does not have a description in the BXML
16	0h	**WDBOX 1 END OF FRAME INTERRUPT (WDBox_1_End_of_Frame_Interrupt):** The field 'WDBox 1 End of Frame Interrupt' in register 'GT 3 Interrupts 0' does not have a description in the BXML
15	0h	**SPARE 15 (Spare_15):** The field 'Spare 15' in register 'GT 3 Interrupts 0' does not have a description in the BXML
14	0h	**SPARE 14 (Spare_14):** The field 'Spare 14' in register 'GT 3 Interrupts 0' does not have a description in the BXML
13	0h	**SPARE 13 (Spare_13):** The field 'Spare 13' in register 'GT 3 Interrupts 0' does not have a description in the BXML
12	0h	**SPARE 12 (Spare_12):** The field 'Spare 12' in register 'GT 3 Interrupts 0' does not have a description in the BXML
11	0h	**VECS WAIT ON SEMAPHORE (VECS_Wait_On_Semaphore):** The field 'VECS Wait On Semaphore' in register 'GT 3 Interrupts 0' does not have a description in the BXML
10	0h	**SPARE 10 (Spare_10):** The field 'Spare 10' in register 'GT 3 Interrupts 0' does not have a description in the BXML
9	0h	**SPARE 9 (Spare_9):** The field 'Spare 9' in register 'GT 3 Interrupts 0' does not have a description in the BXML
8	0h	**VECS CONTEXT SWITCH INTERRUPT (VECS_Context_Switch_Interrupt):** The field 'VECS Context Switch Interrupt' in register 'GT 3 Interrupts 0' does not have a description in the BXML
7	0h	**SPARE 7 (Spare_7):** The field 'Spare 7' in register 'GT 3 Interrupts 0' does not have a description in the BXML
6	0h RO	Reserved.
5	0h	**SPARE 5 (Spare_5):** The field 'Spare 5' in register 'GT 3 Interrupts 0' does not have a description in the BXML
4	0h	**VECS MI FLUSH DW NOTIFY (VECS_MI_Flush_DW_Notify):** The field 'VECS MI Flush DW Notify' in register 'GT 3 Interrupts 0' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
3	0h	**VECS ERROR INTERRUPT (VECS_Error_Interrupt):** The field 'VECS Error Interrupt' in register 'GT 3 Interrupts 0' does not have a description in the BXML
2	0h	**SPARE 2 (Spare_2):** The field 'Spare 2' in register 'GT 3 Interrupts 0' does not have a description in the BXML
1	0h	**SPARE 1 (Spare_1):** The field 'Spare 1' in register 'GT 3 Interrupts 0' does not have a description in the BXML
0	0h	**VECS MI USER INTERRUPT (VECS_MI_User_Interrupt):** The field 'VECS MI User Interrupt' in register 'GT 3 Interrupts 0' does not have a description in the BXML

32.26 GT 3 Interrupts 1 (GT_3_INTERRUPT_1)—Offset 44334h

This table indicates which events are mapped to each bit of the GT Interrupt 3 registers. Bits 15:0 are used for VEBox. Bits 27:16 are used for WDBox. Bits 31:28 are used for OACS. The VEBox Interrupt IIR (sticky) bits are ORed together to generate the VEBox Interrupts Pending bit in the Master Interrupt Control register. The WDBox and OACS Interrupt IIR (sticky) bits are ORed together to generate the WDBox Interrupts Pending bit in the Master Interrupt Control register. 0x44330 = ISR 0x44334 = IMR 0x44338 = IIR 0x4433C = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h	**SPARE 31 (Spare_31):** The field 'Spare 31' in register 'GT 3 Interrupts 1' does not have a description in the BXML
30	0h	**SPARE 30 (Spare_30):** The field 'Spare 30' in register 'GT 3 Interrupts 1' does not have a description in the BXML
29	0h	**SPARE 29 (Spare_29):** The field 'Spare 29' in register 'GT 3 Interrupts 1' does not have a description in the BXML
28	0h	**PERFORMANCE MONITORING BUFFER HALF-FULL INTERRUPT (Performance_Monitoring_Buffer_HalfFull_Interrupt):** For internal trigger (timer event based) reporting, this interrupt is generated if the report buffer crosses the half full limit. For internal trigger (NOA event based) reporting, this interrupt is generated if the report buffer crosses the half full limit.
27	0h	**SPARE 27 (Spare_27):** The field 'Spare 27' in register 'GT 3 Interrupts 1' does not have a description in the BXML
26	0h	**SPARE 26 (Spare_26):** The field 'Spare 26' in register 'GT 3 Interrupts 1' does not have a description in the BXML
25	0h	**SPARE 25 (Spare_25):** The field 'Spare 25' in register 'GT 3 Interrupts 1' does not have a description in the BXML
24	0h	**SPARE 24 (Spare_24):** The field 'Spare 24' in register 'GT 3 Interrupts 1' does not have a description in the BXML
23	0h	**SPARE 23 (Spare_23):** The field 'Spare 23' in register 'GT 3 Interrupts 1' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
22	0h	**SPARE 22 (Spare_22):** The field 'Spare 22' in register 'GT 3 Interrupts 1' does not have a description in the BXML
21	0h	**SPARE 21 (Spare_21):** The field 'Spare 21' in register 'GT 3 Interrupts 1' does not have a description in the BXML
20	0h	**SPARE 20 (Spare_20):** The field 'Spare 20' in register 'GT 3 Interrupts 1' does not have a description in the BXML
19	0h	**SPARE 19 (Spare_19):** The field 'Spare 19' in register 'GT 3 Interrupts 1' does not have a description in the BXML
18	0h	**SPARE 18 (Spare_18):** The field 'Spare 18' in register 'GT 3 Interrupts 1' does not have a description in the BXML
17	0h	**WDBOX 1 STATUS INTERRUPT (WDBox_1_Status_Interrupt):** The field 'WDBox 1 Status Interrupt' in register 'GT 3 Interrupts 1' does not have a description in the BXML
16	0h	**WDBOX 1 END OF FRAME INTERRUPT (WDBox_1_End_of_Frame_Interrupt):** The field 'WDBox 1 End of Frame Interrupt' in register 'GT 3 Interrupts 1' does not have a description in the BXML
15	0h	**SPARE 15 (Spare_15):** The field 'Spare 15' in register 'GT 3 Interrupts 1' does not have a description in the BXML
14	0h	**SPARE 14 (Spare_14):** The field 'Spare 14' in register 'GT 3 Interrupts 1' does not have a description in the BXML
13	0h	**SPARE 13 (Spare_13):** The field 'Spare 13' in register 'GT 3 Interrupts 1' does not have a description in the BXML
12	0h	**SPARE 12 (Spare_12):** The field 'Spare 12' in register 'GT 3 Interrupts 1' does not have a description in the BXML
11	0h	**VECS WAIT ON SEMAPHORE (VECS_Wait_On_Semaphore):** The field 'VECS Wait On Semaphore' in register 'GT 3 Interrupts 1' does not have a description in the BXML
10	0h	**SPARE 10 (Spare_10):** The field 'Spare 10' in register 'GT 3 Interrupts 1' does not have a description in the BXML
9	0h	**SPARE 9 (Spare_9):** The field 'Spare 9' in register 'GT 3 Interrupts 1' does not have a description in the BXML
8	0h	**VECS CONTEXT SWITCH INTERRUPT (VECS_Context_Switch_Interrupt):** The field 'VECS Context Switch Interrupt' in register 'GT 3 Interrupts 1' does not have a description in the BXML
7	0h	**SPARE 7 (Spare_7):** The field 'Spare 7' in register 'GT 3 Interrupts 1' does not have a description in the BXML
6	0h RO	Reserved.
5	0h	**SPARE 5 (Spare_5):** The field 'Spare 5' in register 'GT 3 Interrupts 1' does not have a description in the BXML
4	0h	**VECS MI FLUSH DW NOTIFY (VECS_MI_Flush_DW_Notify):** The field 'VECS MI Flush DW Notify' in register 'GT 3 Interrupts 1' does not have a description in the BXML
3	0h	**VECS ERROR INTERRUPT (VECS_Error_Interrupt):** The field 'VECS Error Interrupt' in register 'GT 3 Interrupts 1' does not have a description in the BXML
2	0h	**SPARE 2 (Spare_2):** The field 'Spare 2' in register 'GT 3 Interrupts 1' does not have a description in the BXML
1	0h	**SPARE 1 (Spare_1):** The field 'Spare 1' in register 'GT 3 Interrupts 1' does not have a description in the BXML
0	0h	**VECS MI USER INTERRUPT (VECS_MI_User_Interrupt):** The field 'VECS MI User Interrupt' in register 'GT 3 Interrupts 1' does not have a description in the BXML

32.27 GT 3 Interrupts 2 (GT_3_INTERRUPT_2)—Offset 44338h

This table indicates which events are mapped to each bit of the GT Interrupt 3 registers. Bits 15:0 are used for VEBox. Bits 27:16 are used for WDBox. Bits 31:28 are used for OACS. The VEBox Interrupt IIR (sticky) bits are ORed together to generate the VEBox Interrupts Pending bit in the Master Interrupt Control register. The WDBox and OACS Interrupt IIR (sticky) bits are ORed together to generate the WDBox Interrupts Pending bit in the Master Interrupt Control register. 0x44330 = ISR 0x44334 = IMR 0x44338 = IIR 0x4433C = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h	**SPARE 31 (Spare_31):** The field 'Spare 31' in register 'GT 3 Interrupts 2' does not have a description in the BXML
30	0h	**SPARE 30 (Spare_30):** The field 'Spare 30' in register 'GT 3 Interrupts 2' does not have a description in the BXML
29	0h	**SPARE 29 (Spare_29):** The field 'Spare 29' in register 'GT 3 Interrupts 2' does not have a description in the BXML
28	0h	**PERFORMANCE MONITORING BUFFER HALF-FULL INTERRUPT (Performance_Monitoring_Buffer_HalfFull_Interrupt):** For internal trigger (timer event based) reporting, this interrupt is generated if the report buffer crosses the half full limit. For internal trigger (NOA event based) reporting, this interrupt is generated if the report buffer crosses the half full limit.
27	0h	**SPARE 27 (Spare_27):** The field 'Spare 27' in register 'GT 3 Interrupts 2' does not have a description in the BXML
26	0h	**SPARE 26 (Spare_26):** The field 'Spare 26' in register 'GT 3 Interrupts 2' does not have a description in the BXML
25	0h	**SPARE 25 (Spare_25):** The field 'Spare 25' in register 'GT 3 Interrupts 2' does not have a description in the BXML
24	0h	**SPARE 24 (Spare_24):** The field 'Spare 24' in register 'GT 3 Interrupts 2' does not have a description in the BXML
23	0h	**SPARE 23 (Spare_23):** The field 'Spare 23' in register 'GT 3 Interrupts 2' does not have a description in the BXML
22	0h	**SPARE 22 (Spare_22):** The field 'Spare 22' in register 'GT 3 Interrupts 2' does not have a description in the BXML
21	0h	**SPARE 21 (Spare_21):** The field 'Spare 21' in register 'GT 3 Interrupts 2' does not have a description in the BXML
20	0h	**SPARE 20 (Spare_20):** The field 'Spare 20' in register 'GT 3 Interrupts 2' does not have a description in the BXML
19	0h	**SPARE 19 (Spare_19):** The field 'Spare 19' in register 'GT 3 Interrupts 2' does not have a description in the BXML
18	0h	**SPARE 18 (Spare_18):** The field 'Spare 18' in register 'GT 3 Interrupts 2' does not have a description in the BXML
17	0h	**WDBOX 1 STATUS INTERRUPT (WDBox_1_Status_Interrupt):** The field 'WDBox 1 Status Interrupt' in register 'GT 3 Interrupts 2' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
16	0h	**WDBOX 1 END OF FRAME INTERRUPT (WDBox_1_End_of_Frame_Interrupt):** The field 'WDBox 1 End of Frame Interrupt' in register 'GT 3 Interrupts 2' does not have a description in the BXML
15	0h	**SPARE 15 (Spare_15):** The field 'Spare 15' in register 'GT 3 Interrupts 2' does not have a description in the BXML
14	0h	**SPARE 14 (Spare_14):** The field 'Spare 14' in register 'GT 3 Interrupts 2' does not have a description in the BXML
13	0h	**SPARE 13 (Spare_13):** The field 'Spare 13' in register 'GT 3 Interrupts 2' does not have a description in the BXML
12	0h	**SPARE 12 (Spare_12):** The field 'Spare 12' in register 'GT 3 Interrupts 2' does not have a description in the BXML
11	0h	**VECS WAIT ON SEMAPHORE (VECS_Wait_On_Semaphore):** The field 'VECS Wait On Semaphore' in register 'GT 3 Interrupts 2' does not have a description in the BXML
10	0h	**SPARE 10 (Spare_10):** The field 'Spare 10' in register 'GT 3 Interrupts 2' does not have a description in the BXML
9	0h	**SPARE 9 (Spare_9):** The field 'Spare 9' in register 'GT 3 Interrupts 2' does not have a description in the BXML
8	0h	**VECS CONTEXT SWITCH INTERRUPT (VECS_Context_Switch_Interrupt):** The field 'VECS Context Switch Interrupt' in register 'GT 3 Interrupts 2' does not have a description in the BXML
7	0h	**SPARE 7 (Spare_7):** The field 'Spare 7' in register 'GT 3 Interrupts 2' does not have a description in the BXML
6	0h RO	Reserved.
5	0h	**SPARE 5 (Spare_5):** The field 'Spare 5' in register 'GT 3 Interrupts 2' does not have a description in the BXML
4	0h	**VECS MI FLUSH DW NOTIFY (VECS_MI_Flush_DW_Notify):** The field 'VECS MI Flush DW Notify' in register 'GT 3 Interrupts 2' does not have a description in the BXML
3	0h	**VECS ERROR INTERRUPT (VECS_Error_Interrupt):** The field 'VECS Error Interrupt' in register 'GT 3 Interrupts 2' does not have a description in the BXML
2	0h	**SPARE 2 (Spare_2):** The field 'Spare 2' in register 'GT 3 Interrupts 2' does not have a description in the BXML
1	0h	**SPARE 1 (Spare_1):** The field 'Spare 1' in register 'GT 3 Interrupts 2' does not have a description in the BXML
0	0h	**VECS MI USER INTERRUPT (VECS_MI_User_Interrupt):** The field 'VECS MI User Interrupt' in register 'GT 3 Interrupts 2' does not have a description in the BXML

32.28 GT 3 Interrupts 3 (GT_3_INTERRUPT_3)—Offset 4433Ch

This table indicates which events are mapped to each bit of the GT Interrupt 3 registers. Bits 15:0 are used for VEBox. Bits 27:16 are used for WDBox. Bits 31:28 are used for OACS. The VEBox Interrupt IIR (sticky) bits are ORed together to generate the VEBox Interrupts Pending bit in the Master Interrupt Control register. The WDBox and OACS Interrupt IIR (sticky) bits are ORed together to generate the WDBox Interrupts Pending bit in the Master Interrupt Control register. 0x44330 = ISR 0x44334 = IMR 0x44338 = IIR 0x4433C = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h	**SPARE 31 (Spare_31):** The field 'Spare 31' in register 'GT 3 Interrupts 3' does not have a description in the BXML
30	0h	**SPARE 30 (Spare_30):** The field 'Spare 30' in register 'GT 3 Interrupts 3' does not have a description in the BXML
29	0h	**SPARE 29 (Spare_29):** The field 'Spare 29' in register 'GT 3 Interrupts 3' does not have a description in the BXML
28	0h	**PERFORMANCE MONITORING BUFFER HALF-FULL INTERRUPT (Performance_Monitoring_Buffer_HalfFull_Interrupt):** For internal trigger (timer event based) reporting, this interrupt is generated if the report buffer crosses the half full limit. For internal trigger (NOA event based) reporting, this interrupt is generated if the report buffer crosses the half full limit.
27	0h	**SPARE 27 (Spare_27):** The field 'Spare 27' in register 'GT 3 Interrupts 3' does not have a description in the BXML
26	0h	**SPARE 26 (Spare_26):** The field 'Spare 26' in register 'GT 3 Interrupts 3' does not have a description in the BXML
25	0h	**SPARE 25 (Spare_25):** The field 'Spare 25' in register 'GT 3 Interrupts 3' does not have a description in the BXML
24	0h	**SPARE 24 (Spare_24):** The field 'Spare 24' in register 'GT 3 Interrupts 3' does not have a description in the BXML
23	0h	**SPARE 23 (Spare_23):** The field 'Spare 23' in register 'GT 3 Interrupts 3' does not have a description in the BXML
22	0h	**SPARE 22 (Spare_22):** The field 'Spare 22' in register 'GT 3 Interrupts 3' does not have a description in the BXML
21	0h	**SPARE 21 (Spare_21):** The field 'Spare 21' in register 'GT 3 Interrupts 3' does not have a description in the BXML
20	0h	**SPARE 20 (Spare_20):** The field 'Spare 20' in register 'GT 3 Interrupts 3' does not have a description in the BXML
19	0h	**SPARE 19 (Spare_19):** The field 'Spare 19' in register 'GT 3 Interrupts 3' does not have a description in the BXML
18	0h	**SPARE 18 (Spare_18):** The field 'Spare 18' in register 'GT 3 Interrupts 3' does not have a description in the BXML
17	0h	**WDBOX 1 STATUS INTERRUPT (WDBox_1_Status_Interrupt):** The field 'WDBox 1 Status Interrupt' in register 'GT 3 Interrupts 3' does not have a description in the BXML
16	0h	**WDBOX 1 END OF FRAME INTERRUPT (WDBox_1_End_of_Frame_Interrupt):** The field 'WDBox 1 End of Frame Interrupt' in register 'GT 3 Interrupts 3' does not have a description in the BXML
15	0h	**SPARE 15 (Spare_15):** The field 'Spare 15' in register 'GT 3 Interrupts 3' does not have a description in the BXML
14	0h	**SPARE 14 (Spare_14):** The field 'Spare 14' in register 'GT 3 Interrupts 3' does not have a description in the BXML
13	0h	**SPARE 13 (Spare_13):** The field 'Spare 13' in register 'GT 3 Interrupts 3' does not have a description in the BXML
12	0h	**SPARE 12 (Spare_12):** The field 'Spare 12' in register 'GT 3 Interrupts 3' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
11	0h	**VECS WAIT ON SEMAPHORE (VECS_Wait_On_Semaphore):** The field 'VECS Wait On Semaphore' in register 'GT 3 Interrupts 3' does not have a description in the BXML
10	0h	**SPARE 10 (Spare_10):** The field 'Spare 10' in register 'GT 3 Interrupts 3' does not have a description in the BXML
9	0h	**SPARE 9 (Spare_9):** The field 'Spare 9' in register 'GT 3 Interrupts 3' does not have a description in the BXML
8	0h	**VECS CONTEXT SWITCH INTERRUPT (VECS_Context_Switch_Interrupt):** The field 'VECS Context Switch Interrupt' in register 'GT 3 Interrupts 3' does not have a description in the BXML
7	0h	**SPARE 7 (Spare_7):** The field 'Spare 7' in register 'GT 3 Interrupts 3' does not have a description in the BXML
6	0h RO	Reserved.
5	0h	**SPARE 5 (Spare_5):** The field 'Spare 5' in register 'GT 3 Interrupts 3' does not have a description in the BXML
4	0h	**VECS MI FLUSH DW NOTIFY (VECS_MI_Flush_DW_Notify):** The field 'VECS MI Flush DW Notify' in register 'GT 3 Interrupts 3' does not have a description in the BXML
3	0h	**VECS ERROR INTERRUPT (VECS_Error_Interrupt):** The field 'VECS Error Interrupt' in register 'GT 3 Interrupts 3' does not have a description in the BXML
2	0h	**SPARE 2 (Spare_2):** The field 'Spare 2' in register 'GT 3 Interrupts 3' does not have a description in the BXML
1	0h	**SPARE 1 (Spare_1):** The field 'Spare 1' in register 'GT 3 Interrupts 3' does not have a description in the BXML
0	0h	**VECS MI USER INTERRUPT (VECS_MI_User_Interrupt):** The field 'VECS MI User Interrupt' in register 'GT 3 Interrupts 3' does not have a description in the BXML

32.29 Display Engine Port Interrupts 0 (DE_PORT_INTERRUPT_0)—Offset 44440h

This table indicates which events are mapped to each bit of the Display Engine Port Interrupt registers. 0x44440 = ISR 0x44444 = IMR 0x44448 = IIR 0x4444C = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**MIPI C (MIPI_C):** The ISR is an active high level indicating an interrupt is set in MIPIC_INTR_STAT_REG or MIPIC_INTR_STAT_REG_1.
30	0h RO	**MIPI A (MIPI_A):** The ISR is an active high level indicating an interrupt is set in MIPIA_INTR_STAT_REG or MIPIA_INTR_STAT_REG_1.
29	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'Display Engine Port Interrupts 0' does not have a description in the BXML
28	0h RO	**AUX CHANNEL F (AUX_Channel_F):** The ISR is an active high pulse on the AUX DDI F done event. This event will not occur for SRD AUX done.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**AUX CHANNEL D (AUX_Channel_D):** The ISR is an active high pulse on the AUX DDI D done event. This event will not occur for SRD AUX done.
26	0h RO	**AUX CHANNEL C (AUX_Channel_C):** The ISR is an active high pulse on the AUX DDI C done event. This event will not occur for SRD AUX done.
25	0h RO	**AUX CHANNEL B (AUX_Channel_B):** The ISR is an active high pulse on the AUX DDI B done event. This event will not occur for SRD AUX done.
24	0h RO	**MIPI C TE (MIPI_C_TE):** The ISR is an active high level indicating a TE interrupt is set in MIPIC_STATUS.
23	0h RO	**MIPI A TE (MIPI_A_TE):** The ISR is an active high level indicating a TE interrupt is set in MIPIA_STATUS.
22:12	0h RSV	**RESERVED (Reserved_1):** The field 'Reserved' in register 'Display Engine Port Interrupts 0' does not have a description in the BXML
11:10	0h RSV	**RESERVED (Reserved_2):** The field 'Reserved' in register 'Display Engine Port Interrupts 0' does not have a description in the BXML
9	0h RO	**SCDC READ REQUEST INTERRUPT - PORT C (SCDC_Read_Request_Interrupt__Port_C):** The IIR is set when a HDMI 2.0 SCDC read request event is detected and is cleared by writing a '1' to this bit. The ISR is active high level signal that will indicate if the read request (RR) is still active.
8	0h RO	**SCDC READ REQUEST INTERRUPT - PORT B (SCDC_Read_Request_Interrupt__Port_B):** The IIR is set when a HDMI 2.0 SCDC read request event is detected and is cleared by writing a '1' to this bit. The ISR is active high level signal that will indicate if the read request (RR) is still active.
7:6	0h RSV	**RESERVED (Reserved_3):** The field 'Reserved' in register 'Display Engine Port Interrupts 0' does not have a description in the BXML
5	0h RO	**DDI C HOTPLUG (DDI_C_Hotplug):** The ISR gives the live state of the DDI HPD pin when the HPD input is enabled. The IIR is set if a short or long pulse is detected when HPD input is enabled. This field is unused in projects that have a PCH.
4	0h RO	**DDI B HOTPLUG (DDI_B_Hotplug):** The ISR gives the live state of the DDI HPD pin when the HPD input is enabled. The IIR is set if a short or long pulse is detected when HPD input is enabled. This field is unused in projects that have a PCH.
3	0h RO	**DDI A HOTPLUG (DDI_A_Hotplug):** The ISR gives the live state of the DDI HPD pin when the HPD input is enabled. The IIR is set if a short or long pulse is detected when HPD input is enabled. This field is unused in projects that have a PCH.
2	0h RSV	**RESERVED (Reserved_4):** The field 'Reserved' in register 'Display Engine Port Interrupts 0' does not have a description in the BXML
1	0h RO	**GMBUS (GMbus):** The ISR is an active high pulse when any of the unmasked events in GMBUS4 Interrupt Mask register occur. This field is only used on projects that have GMBUS integrated into the north display. Projects that have GMBUS in the south display have the GMBUS interrupt in the south display interrupts.
0	0h RO	**AUX_CHANNEL_A (AUX_Channel_A):** The ISR is an active high pulse on the AUX DDI A done event. This event will not occur for SRD AUX done.

32.30 Display Engine Port Interrupts 1 (DE_PORT_INTERRUPT_1)—Offset 44444h

This table indicates which events are mapped to each bit of the Display Engine Port Interrupt registers. 0x44440 = ISR 0x44444 = IMR 0x44448 = IIR 0x4444C = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**MIPI C (MIPI_C):** The ISR is an active high level indicating an interrupt is set in MIPIC_INTR_STAT_REG or MIPIC_INTR_STAT_REG_1.
30	0h RW	**MIPI A (MIPI_A):** The ISR is an active high level indicating an interrupt is set in MIPIA_INTR_STAT_REG or MIPIA_INTR_STAT_REG_1.
29	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'Display Engine Port Interrupts 1' does not have a description in the BXML
28	0h RW	**AUX CHANNEL F (AUX_Channel_F):** The ISR is an active high pulse on the AUX DDI F done event. This event will not occur for SRD AUX done.
27	0h RW	**AUX CHANNEL D (AUX_Channel_D):** The ISR is an active high pulse on the AUX DDI D done event. This event will not occur for SRD AUX done.
26	0h RW	**AUX CHANNEL C (AUX_Channel_C):** The ISR is an active high pulse on the AUX DDI C done event. This event will not occur for SRD AUX done.
25	0h RW	**AUX CHANNEL B (AUX_Channel_B):** The ISR is an active high pulse on the AUX DDI B done event. This event will not occur for SRD AUX done.
24	0h RW	**MIPI C TE (MIPI_C_TE):** The ISR is an active high level indicating a TE interrupt is set in MIPIC_STATUS.
23	0h RW	**MIPI A TE (MIPI_A_TE):** The ISR is an active high level indicating a TE interrupt is set in MIPIA_STATUS.
22:12	0h RSV	**RESERVED (Reserved_1):** The field 'Reserved' in register 'Display Engine Port Interrupts 1' does not have a description in the BXML
11:10	0h RSV	**RESERVED (Reserved_2):** The field 'Reserved' in register 'Display Engine Port Interrupts 1' does not have a description in the BXML
9	0h RW	**SCDC READ REQUEST INTERRUPT - PORT C (SCDC_Read_Request_Interrupt__Port_C):** The IIR is set when a HDMI 2.0 SCDC read request event is detected and is cleared by writing a '1' to this bit. The ISR is active high level signal that will indicate if the read request (RR) is still active.
8	0h RW	**SCDC READ REQUEST INTERRUPT - PORT B (SCDC_Read_Request_Interrupt__Port_B):** The IIR is set when a HDMI 2.0 SCDC read request event is detected and is cleared by writing a '1' to this bit. The ISR is active high level signal that will indicate if the read request (RR) is still active.
7:6	0h RSV	**RESERVED (Reserved_3):** The field 'Reserved' in register 'Display Engine Port Interrupts 1' does not have a description in the BXML
5	0h RW	**DDI C HOTPLUG (DDI_C_Hotplug):** The ISR gives the live state of the DDI HPD pin when the HPD input is enabled. The IIR is set if a short or long pulse is detected when HPD input is enabled. This field is unused in projects that have a PCH.
4	0h RW	**DDI B HOTPLUG (DDI_B_Hotplug):** The ISR gives the live state of the DDI HPD pin when the HPD input is enabled. The IIR is set if a short or long pulse is detected when HPD input is enabled. This field is unused in projects that have a PCH.
3	0h RW	**DDI A HOTPLUG (DDI_A_Hotplug):** The ISR gives the live state of the DDI HPD pin when the HPD input is enabled. The IIR is set if a short or long pulse is detected when HPD input is enabled. This field is unused in projects that have a PCH.
2	0h RSV	**RESERVED (Reserved_4):** The field 'Reserved' in register 'Display Engine Port Interrupts 1' does not have a description in the BXML
1	0h RW	**GMBUS (GMbus):** The ISR is an active high pulse when any of the unmasked events in GMBUS4 Interrupt Mask register occur. This field is only used on projects that have GMBUS integrated into the north display. Projects that have GMBUS in the south display have the GMBUS interrupt in the south display interrupts.
0	0h RW	**AUX_CHANNEL_A (AUX_Channel_A):** The ISR is an active high pulse on the AUX DDI A done event. This event will not occur for SRD AUX done.

32.31 Display Engine Port Interrupts 2 (DE_PORT_INTERRUPT_2)—Offset 44448h

This table indicates which events are mapped to each bit of the Display Engine Port Interrupt registers. 0x44440 = ISR 0x44444 = IMR 0x44448 = IIR 0x4444C = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/C	**MIPI C (MIPI_C):** The ISR is an active high level indicating an interrupt is set in MIPIC_INTR_STAT_REG or MIPIC_INTR_STAT_REG_1.
30	0h RW/C	**MIPI A (MIPI_A):** The ISR is an active high level indicating an interrupt is set in MIPIA_INTR_STAT_REG or MIPIA_INTR_STAT_REG_1.
29	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'Display Engine Port Interrupts 2' does not have a description in the BXML
28	0h RW/C	**AUX CHANNEL F (AUX_Channel_F):** The ISR is an active high pulse on the AUX DDI F done event. This event will not occur for SRD AUX done.
27	0h RW/C	**AUX CHANNEL D (AUX_Channel_D):** The ISR is an active high pulse on the AUX DDI D done event. This event will not occur for SRD AUX done.
26	0h RW/C	**AUX CHANNEL C (AUX_Channel_C):** The ISR is an active high pulse on the AUX DDI C done event. This event will not occur for SRD AUX done.
25	0h RW/C	**AUX CHANNEL B (AUX_Channel_B):** The ISR is an active high pulse on the AUX DDI B done event. This event will not occur for SRD AUX done.
24	0h RW/C	**MIPI C TE (MIPI_C_TE):** The ISR is an active high level indicating a TE interrupt is set in MIPIC_STATUS.
23	0h RW/C	**MIPI A TE (MIPI_A_TE):** The ISR is an active high level indicating a TE interrupt is set in MIPIA_STATUS.
22:12	0h RSV	**RESERVED (Reserved_1):** The field 'Reserved' in register 'Display Engine Port Interrupts 2' does not have a description in the BXML
11:10	0h RSV	**RESERVED (Reserved_2):** The field 'Reserved' in register 'Display Engine Port Interrupts 2' does not have a description in the BXML
9	0h RW/C	**SCDC READ REQUEST INTERRUPT - PORT C (SCDC_Read_Request_Interrupt__Port_C):** The IIR is set when a HDMI 2.0 SCDC read request event is detected and is cleared by writing a '1' to this bit. The ISR is active high level signal that will indicate if the read request (RR) is still active.
8	0h RW/C	**SCDC READ REQUEST INTERRUPT - PORT B (SCDC_Read_Request_Interrupt__Port_B):** The IIR is set when a HDMI 2.0 SCDC read request event is detected and is cleared by writing a '1' to this bit. The ISR is active high level signal that will indicate if the read request (RR) is still active.
7:6	0h RSV	**RESERVED (Reserved_3):** The field 'Reserved' in register 'Display Engine Port Interrupts 2' does not have a description in the BXML
5	0h RW/C	**DDI C HOTPLUG (DDI_C_Hotplug):** The ISR gives the live state of the DDI HPD pin when the HPD input is enabled. The IIR is set if a short or long pulse is detected when HPD input is enabled. This field is unused in projects that have a PCH.
4	0h RW/C	**DDI B HOTPLUG (DDI_B_Hotplug):** The ISR gives the live state of the DDI HPD pin when the HPD input is enabled. The IIR is set if a short or long pulse is detected when HPD input is enabled. This field is unused in projects that have a PCH.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW/C	**DDI A HOTPLUG (DDI_A_Hotplug):** The ISR gives the live state of the DDI HPD pin when the HPD input is enabled. The IIR is set if a short or long pulse is detected when HPD input is enabled. This field is unused in projects that have a PCH.
2	0h RSV	**RESERVED (Reserved_4):** The field 'Reserved' in register 'Display Engine Port Interrupts 2' does not have a description in the BXML
1	0h RW/C	**GMBUS (GMbus):** The ISR is an active high pulse when any of the unmasked events in GMBUS4 Interrupt Mask register occur. This field is only used on projects that have GMBUS integrated into the north display. Projects that have GMBUS in the south display have the GMBUS interrupt in the south display interrupts.
0	0h RW/C	**AUX_CHANNEL_A (AUX_Channel_A):** The ISR is an active high pulse on the AUX DDI A done event. This event will not occur for SRD AUX done.

32.32 Display Engine Port Interrupts 3 (DE_PORT_INTERRUPT_3)—Offset 4444Ch

This table indicates which events are mapped to each bit of the Display Engine Port Interrupt registers. 0x44440 = ISR 0x44444 = IMR 0x44448 = IIR 0x4444C = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**MIPI C (MIPI_C):** The ISR is an active high level indicating an interrupt is set in MIPIC_INTR_STAT_REG or MIPIC_INTR_STAT_REG_1.
30	0h RW	**MIPI A (MIPI_A):** The ISR is an active high level indicating an interrupt is set in MIPIA_INTR_STAT_REG or MIPIA_INTR_STAT_REG_1.
29	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'Display Engine Port Interrupts 3' does not have a description in the BXML
28	0h RW	**AUX CHANNEL F (AUX_Channel_F):** The ISR is an active high pulse on the AUX DDI F done event. This event will not occur for SRD AUX done.
27	0h RW	**AUX CHANNEL D (AUX_Channel_D):** The ISR is an active high pulse on the AUX DDI D done event. This event will not occur for SRD AUX done.
26	0h RW	**AUX CHANNEL C (AUX_Channel_C):** The ISR is an active high pulse on the AUX DDI C done event. This event will not occur for SRD AUX done.
25	0h RW	**AUX CHANNEL B (AUX_Channel_B):** The ISR is an active high pulse on the AUX DDI B done event. This event will not occur for SRD AUX done.
24	0h RW	**MIPI C TE (MIPI_C_TE):** The ISR is an active high level indicating a TE interrupt is set in MIPIC_STATUS.
23	0h RW	**MIPI A TE (MIPI_A_TE):** The ISR is an active high level indicating a TE interrupt is set in MIPIA_STATUS.
22:12	0h RSV	**RESERVED (Reserved_1):** The field 'Reserved' in register 'Display Engine Port Interrupts 3' does not have a description in the BXML
11:10	0h RSV	**RESERVED (Reserved_2):** The field 'Reserved' in register 'Display Engine Port Interrupts 3' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
9	0h RW	**SCDC READ REQUEST INTERRUPT - PORT C (SCDC_Read_Request_Interrupt__Port_C):** The IIR is set when a HDMI 2.0 SCDC read request event is detected and is cleared by writing a '1' to this bit. The ISR is active high level signal that will indicate if the read request (RR) is still active.
8	0h RW	**SCDC READ REQUEST INTERRUPT - PORT B (SCDC_Read_Request_Interrupt__Port_B):** The IIR is set when a HDMI 2.0 SCDC read request event is detected and is cleared by writing a '1' to this bit. The ISR is active high level signal that will indicate if the read request (RR) is still active.
7:6	0h RSV	**RESERVED (Reserved_3):** The field 'Reserved' in register 'Display Engine Port Interrupts 3' does not have a description in the BXML
5	0h RW	**DDI C HOTPLUG (DDI_C_Hotplug):** The ISR gives the live state of the DDI HPD pin when the HPD input is enabled. The IIR is set if a short or long pulse is detected when HPD input is enabled. This field is unused in projects that have a PCH.
4	0h RW	**DDI B HOTPLUG (DDI_B_Hotplug):** The ISR gives the live state of the DDI HPD pin when the HPD input is enabled. The IIR is set if a short or long pulse is detected when HPD input is enabled. This field is unused in projects that have a PCH.
3	0h RW	**DDI A HOTPLUG (DDI_A_Hotplug):** The ISR gives the live state of the DDI HPD pin when the HPD input is enabled. The IIR is set if a short or long pulse is detected when HPD input is enabled. This field is unused in projects that have a PCH.
2	0h RSV	**RESERVED (Reserved_4):** The field 'Reserved' in register 'Display Engine Port Interrupts 3' does not have a description in the BXML
1	0h RW	**GMBUS (GMbus):** The ISR is an active high pulse when any of the unmasked events in GMBUS4 Interrupt Mask register occur. This field is only used on projects that have GMBUS integrated into the north display. Projects that have GMBUS in the south display have the GMBUS interrupt in the south display interrupts.
0	0h RW	**AUX_CHANNEL_A (AUX_Channel_A):** The ISR is an active high pulse on the AUX DDI A done event. This event will not occur for SRD AUX done.

32.33 PCU Interrupts 0 (PCU_INTERRUPT_0)—Offset 444E0h

This table indicates which events are mapped to each bit of the PCU Interrupt registers. 0x444E0 = ISR 0x444E4 = IMR 0x444E8 = IIR 0x444EC = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**DDIA DC9 HPD (DDIA_DC9_HPD):** This field indicates DDIA hotplug activity was detected during DC9.
30	0h RO	**DDIB DC9 HPD (DDIB_DC9_HPD):** This field indicates DDIB hotplug activity was detected during DC9.
29	0h RO	**DDIC DC9 HPD (DDIC_DC9_HPD):** This field indicates DDIC hotplug activity was detected during DC9.
28:26	0h RO	**UNUSED_INT_28_26 (Unused_Int_28_26):** These interrupts are currently unused.
25	0h RO	**PCU_PCODE2DRIVER_MAILBOX_EVENT (PCU_Pcode2driver_Mailbox_Event):** The field 'PCU_Pcode2driver_Mailbox_Event' in register 'PCU Interrupts 0' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
24	0h RO	**PCU_THERMAL_EVENT (PCU_Thermal_Event):** The field 'PCU_Thermal_Event' in register 'PCU Interrupts 0' does not have a description in the BXML
23:0	0h RO	**UNUSED_INT_23_0 (Unused_Int_23_0):** These interrupts are currently unused.

32.34 PCU Interrupts 1 (PCU_INTERRUPT_1)—Offset 444E4h

This table indicates which events are mapped to each bit of the PCU Interrupt registers. 0x444E0 = ISR 0x444E4 = IMR 0x444E8 = IIR 0x444EC = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**DDIA DC9 HPD (DDIA_DC9_HPD):** This field indicates DDIA hotplug activity was detected during DC9.
30	0h RW	**DDIB DC9 HPD (DDIB_DC9_HPD):** This field indicates DDIB hotplug activity was detected during DC9.
29	0h RW	**DDIC DC9 HPD (DDIC_DC9_HPD):** This field indicates DDIC hotplug activity was detected during DC9.
28:26	0h RW	**UNUSED_INT_28_26 (Unused_Int_28_26):** These interrupts are currently unused.
25	0h RW	**PCU_PCODE2DRIVER_MAILBOX_EVENT (PCU_Pcode2driver_Mailbox_Event):** The field 'PCU_Pcode2driver_Mailbox_Event' in register 'PCU Interrupts 1' does not have a description in the BXML
24	0h RW	**PCU_THERMAL_EVENT (PCU_Thermal_Event):** The field 'PCU_Thermal_Event' in register 'PCU Interrupts 1' does not have a description in the BXML
23:0	0h RW	**UNUSED_INT_23_0 (Unused_Int_23_0):** These interrupts are currently unused.

32.35 PCU Interrupts 2 (PCU_INTERRUPT_2)—Offset 444E8h

This table indicates which events are mapped to each bit of the PCU Interrupt registers. 0x444E0 = ISR 0x444E4 = IMR 0x444E8 = IIR 0x444EC = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/C	**DDIA DC9 HPD (DDIA_DC9_HPD):** This field indicates DDIA hotplug activity was detected during DC9.
30	0h RW/C	**DDIB DC9 HPD (DDIB_DC9_HPD):** This field indicates DDIB hotplug activity was detected during DC9.
29	0h RW/C	**DDIC DC9 HPD (DDIC_DC9_HPD):** This field indicates DDIC hotplug activity was detected during DC9.
28:26	0h RW/C	**UNUSED_INT_28_26 (Unused_Int_28_26):** These interrupts are currently unused.
25	0h RW/C	**PCU_PCODE2DRIVER_MAILBOX_EVENT (PCU_Pcode2driver_Mailbox_Event):** The field 'PCU_Pcode2driver_Mailbox_Event' in register 'PCU Interrupts 2' does not have a description in the BXML
24	0h RW/C	**PCU_THERMAL_EVENT (PCU_Thermal_Event):** The field 'PCU_Thermal_Event' in register 'PCU Interrupts 2' does not have a description in the BXML
23:0	0h RW/C	**UNUSED_INT_23_0 (Unused_Int_23_0):** These interrupts are currently unused.

32.36 PCU Interrupts 3 (PCU_INTERRUPT_3)—Offset 444ECh

This table indicates which events are mapped to each bit of the PCU Interrupt registers. 0x444E0 = ISR 0x444E4 = IMR 0x444E8 = IIR 0x444EC = IER

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**DDIA DC9 HPD (DDIA_DC9_HPD):** This field indicates DDIA hotplug activity was detected during DC9.
30	0h RW	**DDIB DC9 HPD (DDIB_DC9_HPD):** This field indicates DDIB hotplug activity was detected during DC9.
29	0h RW	**DDIC DC9 HPD (DDIC_DC9_HPD):** This field indicates DDIC hotplug activity was detected during DC9.
28:26	0h RW	**UNUSED_INT_28_26 (Unused_Int_28_26):** These interrupts are currently unused.
25	0h RW	**PCU_PCODE2DRIVER_MAILBOX_EVENT (PCU_Pcode2driver_Mailbox_Event):** The field 'PCU_Pcode2driver_Mailbox_Event' in register 'PCU Interrupts 3' does not have a description in the BXML
24	0h RW	**PCU_THERMAL_EVENT (PCU_Thermal_Event):** The field 'PCU_Thermal_Event' in register 'PCU Interrupts 3' does not have a description in the BXML
23:0	0h RW	**UNUSED_INT_23_0 (Unused_Int_23_0):** These interrupts are currently unused.

32.37 Display C State Enable (DC_STATE_EN)—Offset 45504h

The register 'Display C State Enable' does not have a description in the BXML source

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**MODE SET IN PROGRESS (MODE_SET_in_Progress):** This bit indicates that Mode set is in progress and DCPR will not generate any CSR_Start to DMC when set. Software needs to program this bit when mode set is started and software should reset it when mode set is done.
30:10	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'Display C State Enable' does not have a description in the BXML
9	0h RW	**IN CSR FLOW (In_CSR_Flow):** Restriction : This field is used for hardware communication. Software must not change this field.
8	0h RW	**BLOCK OUTBOUND TRAFFIC (Block_Outbound_Traffic):** Access is read/write, but hardware can also clear the value based on the PM Request. Restriction : This field is used for hardware communication. Software must not change this field.
7:5	0h RSV	**RESERVED (Reserved_1):** The field 'Reserved' in register 'Display C State Enable' does not have a description in the BXML
4	0h RW	**MASK POKE (Mask_Poke):** This field masks the poke signal that would otherwise be generated by a write to the DC_STATE_SEL register. Restriction : This field is used for hardware communication. Software must not change this field.
3	0h RW	**DC9 ALLOW (DC9_Allow):** This field indicates software allows Display C9. When allowed, the PCU can save the display PCI Config context and power down display
2	0h RSV	**RESERVED (Reserved_2):** The field 'Reserved' in register 'Display C State Enable' does not have a description in the BXML
1:0	0h RW	**DYNAMIC DC STATE ENABLE (Dynamic_DC_State_Enable):** This field enables hardware to dynamically enter and exit Display C states. Restriction : The Display CSR code must be loaded before this field is enabled.

32.38 CD Clock Control (CDCLK_CTL)—Offset 46000h

This register is not reset by the device 2 FLR.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 13Bh

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'CD Clock Control' does not have a description in the BXML
23:22	0h RW	**DE CD2X DIVIDER SELECT (DE_CD2X_Divider_Select):** This field selects how the DE PLL CD2X clock output is divided before driving the display CD2X clock. This field is double buffered to align with the pipe from DE CD2X Pipe Select. It will update at the start of vertical blank of the selected pipe, or immediately if the selected pipe is disabled or no pipe is selected. Restriction : DE CD2X Divider Select must not be changed while more than one pipe is enabled. When one pipe is enabled, the DE CD2X Pipe Select must be set to that pipe before changing DE CD2X Divider Select.

Bit Range	Default & Access	Field Name (ID): Description
21:20	0h RW	**DE CD2X PIPE SELECT (DE_CD2X_Pipe_Select):** This field selects the pipe enable and vertical blank to be used for double buffering the DE CD2X Divider Select and SSA Precharge Enable.
19:18	0h RO	Reserved.
17	0h RSV	**RESERVED (Reserved_1):** The field 'Reserved' in register 'CD Clock Control' does not have a description in the BXML
16	0h RW	**SSA PRECHARGE ENABLE (SSA_Precharge_Enable):** This field enables the Low Speed Small Signal Array Pre-charge. This field is double buffered to align with the pipe from DE CD2X Pipe Select. It will update at the start of vertical blank of the selected pipe, or immediately if the selected pipe is disabled or no pipe is selected. Restriction : This field must be disabled when CD clock frequency is < 500 MHz and enabled when CD clock frequency is >= 500 MHz. This field must not be changed while more than one pipe is enabled. When one pipe is enabled, the DE CD2X Pipe Select must be set to that pipe before changing DE CD2X Divider Select.
15	0h RO	Reserved.
14:11	0h RSV	**RESERVED (Reserved_2):** The field 'Reserved' in register 'CD Clock Control' does not have a description in the BXML
10:0	13Bh RW	**CD FREQUENCY DECIMAL (CD_Frequency_Decimal):** This field selects the decimal value of the frequency for CD clock, which is used to generate divided down clocks for some display engine timers. This value is represented in a 10.1 format with 10 integer bits and 1 fractional bit. Program this field to match the CD frequency chosen by the DE PLL and CD2X Divider, minus one. Used for Aux channel and GMBUS.

32.39 North Display Reset Warn Options (NDE_RSTWRN_OPT)—Offset 46408h

This register is used to control the display behavior on receiving a Reset Warning.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'North Display Reset Warn Options' does not have a description in the BXML
6	0h RSV	**RESERVED (Reserved_1):** The field 'Reserved' in register 'North Display Reset Warn Options' does not have a description in the BXML
5	0h RO	Reserved.
4	0h RW	**RST PCH HANDSHAKE EN (RST_PCH_Handshake_En):** This field enables the handshake with PCH display when processing the reset. This applies to all types of DE resets. By default it is disabled and the north display will not wait for south display to acknowledge the reset. This must be set to 1b as part of the display initialization sequence.
3:0	0h RSV	**RESERVED (Reserved_2):** The field 'Reserved' in register 'North Display Reset Warn Options' does not have a description in the BXML

32.40 Software Flags 0 (SWF_0)—Offset 4F000h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.41 Software Flags 1 (SWF_1)—Offset 4F004h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.42 Software Flags 2 (SWF_2)—Offset 4F008h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.43 Software Flags 3 (SWF_3)—Offset 4F00Ch

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.44 Software Flags 4 (SWF_4)—Offset 4F010h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.45 Software Flags 5 (SWF_5)—Offset 4F014h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.46 Software Flags 6 (SWF_6)—Offset 4F018h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.47 Software Flags 7 (SWF_7)—Offset 4F01Ch

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.48 Software Flags 8 (SWF_8)—Offset 4F020h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.49 Software Flags 9 (SWF_9)—Offset 4F024h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.50 Software Flags 10 (SWF_10)—Offset 4F028h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.51 Software Flags 11 (SWF_11)—Offset 4F02Ch

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.52 Software Flags 12 (SWF_12)—Offset 4F030h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.53 Software Flags 13 (SWF_13)—Offset 4F034h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.54 Software Flags 14 (SWF_14)—Offset 4F038h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.55 Software Flags 15 (SWF_15)—Offset 4F03Ch

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.56 Software Flags 16 (SWF_16)—Offset 4F040h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.57 Software Flags 17 (SWF_17)—Offset 4F044h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.58 Software Flags 18 (SWF_18)—Offset 4F048h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.59 Software Flags 19 (SWF_19)—Offset 4F04Ch

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.60 Software Flags 20 (SWF_20)—Offset 4F050h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.61 Software Flags 21 (SWF_21)—Offset 4F054h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.62 Software Flags 22 (SWF_22)—Offset 4F058h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.63 Software Flags 23 (SWF_23)—Offset 4F05Ch

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.64 Software Flags 24 (SWF_24)—Offset 4F060h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.65 Software Flags 25 (SWF_25)—Offset 4F064h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.66 Software Flags 26 (SWF_26)—Offset 4F068h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.67 Software Flags 27 (SWF_27)—Offset 4F06Ch

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.68 Software Flags 28 (SWF_28)—Offset 4F070h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.69 Software Flags 29 (SWF_29)—Offset 4F074h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.70 Software Flags 30 (SWF_30)—Offset 4F078h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.71 Software Flags 31 (SWF_31)—Offset 4F07Ch

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.72 Software Flags 32 (SWF_32)—Offset 4F080h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.73 Software Flags 33 (SWF_33)—Offset 4F084h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.74 Software Flags 34 (SWF_34)—Offset 4F088h

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.75 Software Flags 35 (SWF_35)—Offset 4F08Ch

These registers are used as scratch pad data storage space and have no direct effect on hardware operation. There are 36 instances of this register format.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**SOFTWARE FLAGS (Software_Flags):** Software flags

32.76 Display Fuse (DFSM)—Offset 51000h

This register contains fuse and strap settings for display. This register is not reset by a debug reset or FLR. Access is RO_FW. Only writes from the correct source will update the register value. Writes from other sources will complete normally without updating the value. Any source can read the register.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: B00h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**SPARE 31 (Spare_31):** The field 'Spare 31' in register 'Display Fuse' does not have a description in the BXML
30	0h RW	**DISPLAY PIPEA DISABLE (Display_PipeA_Disable):** This bit indicates whether the display pipe A (first pipe) capability is disabled. When the fuse is set, display hardware will override the pipe A output to a solid color.
29	0h RO	Reserved.
28	0h RW	**DISPLAY PIPEC DISABLE (Display_PipeC_Disable):** This bit indicates whether the display pipe C (third pipe) capability is disabled. When the fuse is set, display hardware will override the pipe C output to a solid color.
27	0h RW	**DISPLAY PM DISABLE (Display_PM_Disable):** This bit indicates whether the display power management FBC and DPST capabilities are disabled. When disabled, display hardware will prevent the FBC enable and DPST image enhancement enable register bits from being set to 1b.
26	0h RW	**DISPLAY EDP DISABLE (Display_eDP_Disable):** This bit indicates whether the display embedded DisplayPort eDP DDIA capability is disabled. When disabled, display hardware will prevent the eDP DDIA enable register bit from being set to 1b and mask the eDP DDIA present strap.
25	0h RO	Reserved.
24	0h RW	**SPARE 24 (Spare_24):** The field 'Spare 24' in register 'Display Fuse' does not have a description in the BXML
23	0h RW	**SPARE 23 (Spare_23):** The field 'Spare 23' in register 'Display Fuse' does not have a description in the BXML
22	0h RW	**SPARE 22 (Spare_22):** The field 'Spare 22' in register 'Display Fuse' does not have a description in the BXML
21	0h RW	**DISPLAY PIPEB DISABLE (Display_PipeB_Disable):** This bit indicates whether the display pipe B (second pipe) capability is disabled. When the fuse is set, display hardware will override the pipe B output to a solid color.
20	0h RW	**DISPLAY WD DISABLE (Display_WD_Disable):** This bit indicates whether the display WD Video capability is disabled. When disabled, display hardware will prevent the pipe WD Video function enable register bit from being set to 1b. WD audio is not impacted by this fuse.
19	0h RW	**SPARE 19 (Spare_19)**
18	0h RW	**SPARE 18 (Spare_18):** The field 'Spare 18' in register 'Display Fuse' does not have a description in the BXML
17	0h RW	**SPARE 17 (Spare_17):** The field 'Spare 17' in register 'Display Fuse' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
16	0h RW	**ISOLATED DECODE DISABLE (Isolated_Decode_Disable):** This field indicates whether the Isolated Decode feature is disabled.
15:8	Bh RW	**AUDIO CODEC ID (Audio_Codec_ID):** This field indicates the lower 8 bits of the audio codec device ID.
7	0h RW	**DISPLAY DSC DISABLE (Display_DSC_Disable):** This field indicates whether the DSC (MIPI DSI and eDP compression) feature is disabled.
6	0h RW	**DISPLAY RSB ENABLE (Display_RSB_Enable):** This bit indicates whether the remote screen blanking feature is enabled in the display engine. When disabled, display hardware will prevent the remote screen blanking from being enabled.
5	0h RW	**SPARE 5 (Spare_5):** The field 'Spare 5' in register 'Display Fuse' does not have a description in the BXML
4	0h RW	**SPARE 4 (Spare_4):** The field 'Spare 4' in register 'Display Fuse' does not have a description in the BXML
3	0h RW	**SPARE 3 (Spare_3):** The field 'Spare 3' in register 'Display Fuse' does not have a description in the BXML
2	0h RW	**SPARE 2 (Spare_2):** The field 'Spare 2' in register 'Display Fuse' does not have a description in the BXML
1	0h RO	Reserved.
0	0h RW	**DISPLAY AUDIO CODEC DISABLE (Display_Audio_Codec_Disable):** This bit indicates whether the display audio codec capability is disabled. When disabled, display hardware will prevent the audio codec enable register bit from being set to 1b.

32.77 Display Strap State (DSSM)—Offset 51004h

This register contains fuse and strap settings for display. This register is not reset by a debug reset or FLR. Access is RO_FW. Only writes from cfgspace 0x11111 and srcID 0x10 will update the register value. Writes from other sources will complete normally without updating the value. Any source can read the register.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO	**REFERENCE FREQUENCY (Reference_Frequency):** This bit indicates the reference clock frequency. Software should use this value when programming the display clocks.
30	0h RW	**SPARE 30 (Spare_30):** The field 'Spare 30' in register 'Display Strap State' does not have a description in the BXML
29	0h RW	**SPARE 29 (Spare_29):** The field 'Spare 29' in register 'Display Strap State' does not have a description in the BXML
28	0h RW	**SPARE 28 (Spare_28):** The field 'Spare 28' in register 'Display Strap State' does not have a description in the BXML
27	0h RW	**SPARE 27 (Spare_27):** The field 'Spare 27' in register 'Display Strap State' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
26	0h RW	**SPARE 26 (Spare_26):** The field 'Spare 26' in register 'Display Strap State' does not have a description in the BXML
25	0h RW	**SPARE 25 (Spare_25):** The field 'Spare 25' in register 'Display Strap State' does not have a description in the BXML
24	0h RW	**SPARE 24 (Spare_24):** The field 'Spare 24' in register 'Display Strap State' does not have a description in the BXML
23	0h RW	**SPARE 23 (Spare_23):** The field 'Spare 23' in register 'Display Strap State' does not have a description in the BXML
22	0h RW	**SPARE 22 (Spare_22):** The field 'Spare 22' in register 'Display Strap State' does not have a description in the BXML
21	0h RW	**SPARE 21 (Spare_21):** The field 'Spare 21' in register 'Display Strap State' does not have a description in the BXML
20	0h RW	**SPARE 20 (Spare_20):** The field 'Spare 20' in register 'Display Strap State' does not have a description in the BXML
19	0h RW	**SPARE 19 (Spare_19):** The field 'Spare 19' in register 'Display Strap State' does not have a description in the BXML
18	0h RW	**SPARE 18 (Spare_18):** The field 'Spare 18' in register 'Display Strap State' does not have a description in the BXML
17	0h RW	**SPARE 17 (Spare_17):** The field 'Spare 17' in register 'Display Strap State' does not have a description in the BXML
16	0h RW	**SPARE 16 (Spare_16):** The field 'Spare 16' in register 'Display Strap State' does not have a description in the BXML
15	0h RW	**SPARE 15 (Spare_15):** The field 'Spare 15' in register 'Display Strap State' does not have a description in the BXML
14	0h RW	**SPARE 14 (Spare_14):** The field 'Spare 14' in register 'Display Strap State' does not have a description in the BXML
13	0h RW	**SPARE 13 (Spare_13):** The field 'Spare 13' in register 'Display Strap State' does not have a description in the BXML
12	0h RW	**SPARE 12 (Spare_12):** The field 'Spare 12' in register 'Display Strap State' does not have a description in the BXML
11	0h RW	**SPARE 11 (Spare_11):** The field 'Spare 11' in register 'Display Strap State' does not have a description in the BXML
10	0h RW	**SPARE 10 (Spare_10):** The field 'Spare 10' in register 'Display Strap State' does not have a description in the BXML
9	0h RW	**SPARE 9 (Spare_9):** The field 'Spare 9' in register 'Display Strap State' does not have a description in the BXML
8	0h RW	**SPARE 8 (Spare_8):** The field 'Spare 8' in register 'Display Strap State' does not have a description in the BXML
7	0h RW	**SPARE 7 (Spare_7):** The field 'Spare 7' in register 'Display Strap State' does not have a description in the BXML
6	0h RW	**SPARE 6 (Spare_6):** The field 'Spare 6' in register 'Display Strap State' does not have a description in the BXML
5	0h RW	**SPARE 5 (Spare_5):** The field 'Spare 5' in register 'Display Strap State' does not have a description in the BXML
4	0h RW	**SPARE 4 (Spare_4):** The field 'Spare 4' in register 'Display Strap State' does not have a description in the BXML
3	0h RW	**SPARE 3 (Spare_3):** The field 'Spare 3' in register 'Display Strap State' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
2	0h RW	**SPARE 2 (Spare_2):** The field 'Spare 2' in register 'Display Strap State' does not have a description in the BXML
1	0h RW	**PART IS SOC (Part_Is_SOC):** This field specifies whether this part is a SoC or not.
0	0h RW	**DISPLAYPORT A PRESENT (DisplayPort_A_Present):** This bit specifies whether the port was present during initalization. This strap state can also be read in the DDI_BUF_CTL_A 0x64000 register bit 0. There are no port presence straps on Broxton. Software should use alternate means to determine port presence. Workaround : The strap is not connected on the A steppings.

32.78 Display Fuse Done (DFSDONE)—Offset 51080h

This register is not reset by a debug reset or FLR. Access is RO_FW. Only writes from PCU (cfgspace 0x11111 and srcID 0x10) to Display on the message channel will update the register value. Writes from other sources will complete normally without updating the value. Any source can read the register.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'Display Fuse Done' does not have a description in the BXML
0	0h RW	**DOWNLOAD DONE (Download_Done):** This field indicates when fuse download is complete.

32.79 DDI A Buffer Control (DDI_BUF_CTL_A)—Offset 64000h

There is one DDI Buffer Control per each DDI A/B/C/D/E/F. Do not read or write the register when the associated power well is disabled.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**DDI BUFFER ENABLE (DDI_Buffer_Enable):** This bit enables the DDI buffer.
30	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'DDI A Buffer Control' does not have a description in the BXML

Bit Range	Default & Access	Field Name (ID): Description
29	0h RW	**OVERRIDE TRAINING ENABLE (Override_Training_Enable):** This field enables the override on the training enable signal that tells the DDI I/O to pick up any DDI voltage swing and pre-emphasis changes.
28	0h RW	**PHY PARAM ADJUST (Phy_Param_Adjust):** Enables adjustment of Phy parameters such as voltage swing and pre emphasis outside lnik training process. This field is conditioned on 'override training enable' (DDI_BUF_CTL[29]).
27:24	0h RSV	**RESERVED (Reserved_1):** The field 'Reserved' in register 'DDI A Buffer Control' does not have a description in the BXML
23:17	0h RSV	**RESERVED (Reserved_2):** The field 'Reserved' in register 'DDI A Buffer Control' does not have a description in the BXML
16	0h RW	**PORT REVERSAL (Port_Reversal):** This field enables lane reversal within the port. Lane reversal swaps the data on the lanes as they are output from the port. DDI B, C, D, and F reversal always swaps the four lanes, so lane 0 is swapped with lane 3, and lane 1 is swapped with lane 2. If DDIA Lane Capability Control selects DDIA x2, then DDI A reversal swaps the two lanes, so lane 0 is swapped with lane 1. If DDIA Lane Capability Control selects DDIA x4, then DDI A reversal swaps the four lanes, so lane 0 is swapped with lane 3, and lane 1 is swapped with lane 2. Restriction : This field must not be changed while the DDI is enabled. DDI E does not support reversal.
15:8	0h RSV	**RESERVED (Reserved_3):** The field 'Reserved' in register 'DDI A Buffer Control' does not have a description in the BXML
7	0h RO	**DDI IDLE STATUS (DDI_Idle_Status):** This bit indicates when the DDI buffer is idle.
6:5	0h RSV	**RESERVED (Reserved_4):** The field 'Reserved' in register 'DDI A Buffer Control' does not have a description in the BXML
4	0h RW	**DDIA LANE CAPABILITY CONTROL (DDIA_Lane_Capability_Control):** This bit selects how lanes are shared between DDI A and DDI E. This field is only used in the DDI A instance of this register. See the DDI A and DDI E lane mapping table in the Introduction section. Restriction : This field must be programmed at system boot based on board configuration and may not be changed afterwards. Restriction : DDI A x2 (DDI A x2 + DDI E x2) selection is not valid on boards that use DDI F.
3:1	0h RW	**DP PORT WIDTH SELECTION (DP_Port_Width_Selection):** This bit selects the number of lanes to be enabled on the DDI link for DisplayPort. This field is ignored for HDMI and DVI which always use all 4 lanes. Restriction : When in DisplayPort mode the value selected here must match the value selected in the DDI Buffer Control register for the DDI attached to this pipe. Restriction : This field must not be changed while the DDI is enabled. DDI E only supports x1 and x2 when DDI_BUF_CTL_A DDIA Lane Capability Control is set to DDIA x2, otherwise DDI E is not supported. DDI A (EDP) supports x1, x2, and x4 when DDI_BUF_CTL_A DDIA Lane Capability Control is set to DDIA x4, otherwise DDI A only supports x1 and x2.
0	0h RO	**INIT DISPLAY DETECTED (Init_Display_Detected):** Strap indicating whether a display was detected on this port during initialization. It signifies the level of the port detect pin at boot. This bit is only informative. It does not prevent this port from being enabled in hardware. This field only indicates the DDIA detection. Detection for other ports is read from SFUSE_STRAP.

32.80 DDI B Buffer Control (DDI_BUF_CTL_B)—Offset 64100h

There is one DDI Buffer Control per each DDI A/B/C/D/E/F. Do not read or write the register when the associated power well is disabled.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**DDI BUFFER ENABLE (DDI_Buffer_Enable):** This bit enables the DDI buffer.
30	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'DDI B Buffer Control' does not have a description in the BXML
29	0h RW	**OVERRIDE TRAINING ENABLE (Override_Training_Enable):** This field enables the override on the training enable signal that tells the DDI I/O to pick up any DDI voltage swing and pre-emphasis changes.
28	0h RW	**PHY PARAM ADJUST (Phy_Param_Adjust):** Enables adjustment of Phy parameters such as voltage swing and pre emphasis outside lnik training process. This field is conditioned on 'override training enable' (DDI_BUF_CTL[29]).
27:24	0h RSV	**RESERVED (Reserved_1):** The field 'Reserved' in register 'DDI B Buffer Control' does not have a description in the BXML
23:17	0h RSV	**RESERVED (Reserved_2):** The field 'Reserved' in register 'DDI B Buffer Control' does not have a description in the BXML
16	0h RW	**PORT REVERSAL (Port_Reversal):** This field enables lane reversal within the port. Lane reversal swaps the data on the lanes as they are output from the port. DDI B, C, D, and F reversal always swaps the four lanes, so lane 0 is swapped with lane 3, and lane 1 is swapped with lane 2. If DDIA Lane Capability Control selects DDIA x2, then DDI A reversal swaps the two lanes, so lane 0 is swapped with lane 1. If DDIA Lane Capability Control selects DDIA x4, then DDI A reversal swaps the four lanes, so lane 0 is swapped with lane 3, and lane 1 is swapped with lane 2. Restriction : This field must not be changed while the DDI is enabled. DDI E does not support reversal.
15:8	0h RSV	**RESERVED (Reserved_3):** The field 'Reserved' in register 'DDI B Buffer Control' does not have a description in the BXML
7	0h RO	**DDI IDLE STATUS (DDI_Idle_Status):** This bit indicates when the DDI buffer is idle.
6:5	0h RSV	**RESERVED (Reserved_4):** The field 'Reserved' in register 'DDI B Buffer Control' does not have a description in the BXML
4	0h RW	**DDIA LANE CAPABILITY CONTROL (DDIA_Lane_Capability_Control):** This bit selects how lanes are shared between DDI A and DDI E. This field is only used in the DDI A instance of this register. See the DDI A and DDI E lane mapping table in the Introduction section. Restriction : This field must be programmed at system boot based on board configuration and may not be changed afterwards. Restriction : DDI A x2 (DDI A x2 + DDI E x2) selection is not valid on boards that use DDI F.
3:1	0h RW	**DP PORT WIDTH SELECTION (DP_Port_Width_Selection):** This bit selects the number of lanes to be enabled on the DDI link for DisplayPort. This field is ignored for HDMI and DVI which always use all 4 lanes. Restriction : When in DisplayPort mode the value selected here must match the value selected in the DDI Buffer Control register for the DDI attached to this pipe. Restriction : This field must not be changed while the DDI is enabled. DDI E only supports x1 and x2 when DDI_BUF_CTL_A DDIA Lane Capability Control is set to DDIA x2, otherwise DDI E is not supported. DDI A (EDP) supports x1, x2, and x4 when DDI_BUF_CTL_A DDIA Lane Capability Control is set to DDIA x4, otherwise DDI A only supports x1 and x2.
0	0h RO	**INIT DISPLAY DETECTED (Init_Display_Detected):** Strap indicating whether a display was detected on this port during initialization. It signifies the level of the port detect pin at boot. This bit is only informative. It does not prevent this port from being enabled in hardware. This field only indicates the DDIA detection. Detection for other ports is read from SFUSE_STRAP.

32.81 DDI C Buffer Control (DDI_BUF_CTL_C)—Offset 64200h

There is one DDI Buffer Control per each DDI A/B/C/D/E/F. Do not read or write the register when the associated power well is disabled.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**DDI BUFFER ENABLE (DDI_Buffer_Enable):** This bit enables the DDI buffer.
30	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'DDI C Buffer Control' does not have a description in the BXML
29	0h RW	**OVERRIDE TRAINING ENABLE (Override_Training_Enable):** This field enables the override on the training enable signal that tells the DDI I/O to pick up any DDI voltage swing and pre-emphasis changes.
28	0h RW	**PHY PARAM ADJUST (Phy_Param_Adjust):** Enables adjustment of Phy parameters such as voltage swing and pre emphasis outside lnik training process. This field is conditioned on 'override training enable' (DDI_BUF_CTL[29]).
27:24	0h RSV	**RESERVED (Reserved_1):** The field 'Reserved' in register 'DDI C Buffer Control' does not have a description in the BXML
23:17	0h RSV	**RESERVED (Reserved_2):** The field 'Reserved' in register 'DDI C Buffer Control' does not have a description in the BXML
16	0h RW	**PORT REVERSAL (Port_Reversal):** This field enables lane reversal within the port. Lane reversal swaps the data on the lanes as they are output from the port. DDI B, C, D, and F reversal always swaps the four lanes, so lane 0 is swapped with lane 3, and lane 1 is swapped with lane 2. If DDIA Lane Capability Control selects DDIA x2, then DDI A reversal swaps the two lanes, so lane 0 is swapped with lane 1. If DDIA Lane Capability Control selects DDIA x4, then DDI A reversal swaps the four lanes, so lane 0 is swapped with lane 3, and lane 1 is swapped with lane 2. Restriction : This field must not be changed while the DDI is enabled. DDI E does not support reversal.
15:8	0h RSV	**RESERVED (Reserved_3):** The field 'Reserved' in register 'DDI C Buffer Control' does not have a description in the BXML
7	0h RO	**DDI IDLE STATUS (DDI_Idle_Status):** This bit indicates when the DDI buffer is idle.
6:5	0h RSV	**RESERVED (Reserved_4):** The field 'Reserved' in register 'DDI C Buffer Control' does not have a description in the BXML
4	0h RW	**DDIA LANE CAPABILITY CONTROL (DDIA_Lane_Capability_Control):** This bit selects how lanes are shared between DDI A and DDI E. This field is only used in the DDI A instance of this register. See the DDI A and DDI E lane mapping table in the Introduction section. Restriction : This field must be programmed at system boot based on board configuration and may not be changed afterwards. Restriction : DDI A x2 (DDI A x2 + DDI E x2) selection is not valid on boards that use DDI F.
3:1	0h RW	**DP PORT WIDTH SELECTION (DP_Port_Width_Selection):** This bit selects the number of lanes to be enabled on the DDI link for DisplayPort. This field is ignored for HDMI and DVI which always use all 4 lanes. Restriction : When in DisplayPort mode the value selected here must match the value selected in the DDI Buffer Control register for the DDI attached to this pipe. Restriction : This field must not be changed while the DDI is enabled. DDI E only supports x1 and x2 when DDI_BUF_CTL_A DDIA Lane Capability Control is set to DDIA x2, otherwise DDI E is not supported. DDI A (EDP) supports x1, x2, and x4 when DDI_BUF_CTL_A DDIA Lane Capability Control is set to DDIA x4, otherwise DDI A only supports x1 and x2.
0	0h RO	**INIT DISPLAY DETECTED (Init_Display_Detected):** Strap indicating whether a display was detected on this port during initialization. It signifies the level of the port detect pin at boot. This bit is only informative. It does not prevent this port from being enabled in hardware. This field only indicates the DDIA detection. Detection for other ports is read from SFUSE_STRAP.

32.82 DDI D Buffer Control (DDI_BUF_CTL_D)—Offset 64300h

There is one DDI Buffer Control per each DDI A/B/C/D/E/F. Do not read or write the register when the associated power well is disabled.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**DDI BUFFER ENABLE (DDI_Buffer_Enable):** This bit enables the DDI buffer.
30	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'DDI D Buffer Control' does not have a description in the BXML
29	0h RW	**OVERRIDE TRAINING ENABLE (Override_Training_Enable):** This field enables the override on the training enable signal that tells the DDI I/O to pick up any DDI voltage swing and pre-emphasis changes.
28	0h RW	**PHY PARAM ADJUST (Phy_Param_Adjust):** Enables adjustment of Phy parameters such as voltage swing and pre emphasis outside lnik training process. This field is conditioned on 'override training enable' (DDI_BUF_CTL[29]).
27:24	0h RSV	**RESERVED (Reserved_1):** The field 'Reserved' in register 'DDI D Buffer Control' does not have a description in the BXML
23:17	0h RSV	**RESERVED (Reserved_2):** The field 'Reserved' in register 'DDI D Buffer Control' does not have a description in the BXML
16	0h RW	**PORT REVERSAL (Port_Reversal):** This field enables lane reversal within the port. Lane reversal swaps the data on the lanes as they are output from the port. DDI B, C, D, and F reversal always swaps the four lanes, so lane 0 is swapped with lane 3, and lane 1 is swapped with lane 2. If DDIA Lane Capability Control selects DDIA x2, then DDI A reversal swaps the two lanes, so lane 0 is swapped with lane 1. If DDIA Lane Capability Control selects DDIA x4, then DDI A reversal swaps the four lanes, so lane 0 is swapped with lane 3, and lane 1 is swapped with lane 2. Restriction : This field must not be changed while the DDI is enabled. DDI E does not support reversal.
15:8	0h RSV	**RESERVED (Reserved_3):** The field 'Reserved' in register 'DDI D Buffer Control' does not have a description in the BXML
7	0h RO	**DDI IDLE STATUS (DDI_Idle_Status):** This bit indicates when the DDI buffer is idle.
6:5	0h RSV	**RESERVED (Reserved_4):** The field 'Reserved' in register 'DDI D Buffer Control' does not have a description in the BXML
4	0h RW	**DDIA LANE CAPABILITY CONTROL (DDIA_Lane_Capability_Control):** This bit selects how lanes are shared between DDI A and DDI E. This field is only used in the DDI A instance of this register. See the DDI A and DDI E lane mapping table in the Introduction section. Restriction : This field must be programmed at system boot based on board configuration and may not be changed afterwards. Restriction : DDI A x2 (DDI A x2 + DDI E x2) selection is not valid on boards that use DDI F.
3:1	0h RW	**DP PORT WIDTH SELECTION (DP_Port_Width_Selection):** This bit selects the number of lanes to be enabled on the DDI link for DisplayPort. This field is ignored for HDMI and DVI which always use all 4 lanes. Restriction : When in DisplayPort mode the value selected here must match the value selected in the DDI Buffer Control register for the DDI attached to this pipe. Restriction : This field must not be changed while the DDI is enabled. DDI E only supports x1 and x2 when DDI_BUF_CTL_A DDIA Lane Capability Control is set to DDIA x2, otherwise DDI E is not supported. DDI A (EDP) supports x1, x2, and x4 when DDI_BUF_CTL_A DDIA Lane Capability Control is set to DDIA x4, otherwise DDI A only supports x1 and x2.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**INIT DISPLAY DETECTED (Init_Display_Detected):** Strap indicating whether a display was detected on this port during initialization. It signifies the level of the port detect pin at boot. This bit is only informative. It does not prevent this port from being enabled in hardware. This field only indicates the DDIA detection. Detection for other ports is read from SFUSE_STRAP.

32.83 DDI E Buffer Control (DDI_BUF_CTL_E)—Offset 64400h

There is one DDI Buffer Control per each DDI A/B/C/D/E/F. Do not read or write the register when the associated power well is disabled.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**DDI BUFFER ENABLE (DDI_Buffer_Enable):** This bit enables the DDI buffer.
30	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'DDI E Buffer Control' does not have a description in the BXML
29	0h RW	**OVERRIDE TRAINING ENABLE (Override_Training_Enable):** This field enables the override on the training enable signal that tells the DDI I/O to pick up any DDI voltage swing and pre-emphasis changes.
28	0h RW	**PHY PARAM ADJUST (Phy_Param_Adjust):** Enables adjustment of Phy parameters such as voltage swing and pre emphasis outside Inik training process. This field is conditioned on 'override training enable' (DDI_BUF_CTL[29]).
27:24	0h RSV	**RESERVED (Reserved_1):** The field 'Reserved' in register 'DDI E Buffer Control' does not have a description in the BXML
23:17	0h RSV	**RESERVED (Reserved_2):** The field 'Reserved' in register 'DDI E Buffer Control' does not have a description in the BXML
16	0h RW	**PORT REVERSAL (Port_Reversal):** This field enables lane reversal within the port. Lane reversal swaps the data on the lanes as they are output from the port. DDI B, C, D, and F reversal always swaps the four lanes, so lane 0 is swapped with lane 3, and lane 1 is swapped with lane 2. If DDIA Lane Capability Control selects DDIA x2, then DDI A reversal swaps the two lanes, so lane 0 is swapped with lane 1. If DDIA Lane Capability Control selects DDIA x4, then DDI A reversal swaps the four lanes, so lane 0 is swapped with lane 3, and lane 1 is swapped with lane 2. Restriction : This field must not be changed while the DDI is enabled. DDI E does not support reversal.
15:8	0h RSV	**RESERVED (Reserved_3):** The field 'Reserved' in register 'DDI E Buffer Control' does not have a description in the BXML
7	0h RO	**DDI IDLE STATUS (DDI_Idle_Status):** This bit indicates when the DDI buffer is idle.
6:5	0h RSV	**RESERVED (Reserved_4):** The field 'Reserved' in register 'DDI E Buffer Control' does not have a description in the BXML
4	0h RW	**DDIA LANE CAPABILITY CONTROL (DDIA_Lane_Capability_Control):** This bit selects how lanes are shared between DDI A and DDI E. This field is only used in the DDI A instance of this register. See the DDI A and DDI E lane mapping table in the Introduction section. Restriction : This field must be programmed at system boot based on board configuration and may not be changed afterwards. Restriction : DDI A x2 (DDI A x2 + DDI E x2) selection is not valid on boards that use DDI F.

Bit Range	Default & Access	Field Name (ID): Description
3:1	0h RW	**DP PORT WIDTH SELECTION (DP_Port_Width_Selection):** This bit selects the number of lanes to be enabled on the DDI link for DisplayPort. This field is ignored for HDMI and DVI which always use all 4 lanes. Restriction : When in DisplayPort mode the value selected here must match the value selected in the DDI Buffer Control register for the DDI attached to this pipe. Restriction : This field must not be changed while the DDI is enabled. DDI E only supports x1 and x2 when DDI_BUF_CTL_A DDIA Lane Capability Control is set to DDIA x2, otherwise DDI E is not supported. DDI A (EDP) supports x1, x2, and x4 when DDI_BUF_CTL_A DDIA Lane Capability Control is set to DDIA x4, otherwise DDI A only supports x1 and x2.
0	0h RO	**INIT DISPLAY DETECTED (Init_Display_Detected):** Strap indicating whether a display was detected on this port during initialization. It signifies the level of the port detect pin at boot. This bit is only informative. It does not prevent this port from being enabled in hardware. This field only indicates the DDIA detection. Detection for other ports is read from SFUSE_STRAP.

32.84 DDI F Buffer Control (DDI_BUF_CTL_F)—Offset 64500h

There is one DDI Buffer Control per each DDI A/B/C/D/E/F. Do not read or write the register when the associated power well is disabled.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**DDI BUFFER ENABLE (DDI_Buffer_Enable):** This bit enables the DDI buffer.
30	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'DDI F Buffer Control' does not have a description in the BXML
29	0h RW	**OVERRIDE TRAINING ENABLE (Override_Training_Enable):** This field enables the override on the training enable signal that tells the DDI I/O to pick up any DDI voltage swing and pre-emphasis changes.
28	0h RW	**PHY PARAM ADJUST (Phy_Param_Adjust):** Enables adjustment of Phy parameters such as voltage swing and pre emphasis outside lnik training process. This field is conditioned on 'override training enable' (DDI_BUF_CTL[29]).
27:24	0h RSV	**RESERVED (Reserved_1):** The field 'Reserved' in register 'DDI F Buffer Control' does not have a description in the BXML
23:17	0h RSV	**RESERVED (Reserved_2):** The field 'Reserved' in register 'DDI F Buffer Control' does not have a description in the BXML
16	0h RW	**PORT REVERSAL (Port_Reversal):** This field enables lane reversal within the port. Lane reversal swaps the data on the lanes as they are output from the port. DDI B, C, D, and F reversal always swaps the four lanes, so lane 0 is swapped with lane 3, and lane 1 is swapped with lane 2. If DDIA Lane Capability Control selects DDIA x2, then DDI A reversal swaps the two lanes, so lane 0 is swapped with lane 1. If DDIA Lane Capability Control selects DDIA x4, then DDI A reversal swaps the four lanes, so lane 0 is swapped with lane 3, and lane 1 is swapped with lane 2. Restriction : This field must not be changed while the DDI is enabled. DDI E does not support reversal.
15:8	0h RSV	**RESERVED (Reserved_3):** The field 'Reserved' in register 'DDI F Buffer Control' does not have a description in the BXML
7	0h RO	**DDI IDLE STATUS (DDI_Idle_Status):** This bit indicates when the DDI buffer is idle.

Bit Range	Default & Access	Field Name (ID): Description
6:5	0h RSV	**RESERVED (Reserved_4):** The field 'Reserved' in register 'DDI F Buffer Control' does not have a description in the BXML
4	0h RW	**DDIA LANE CAPABILITY CONTROL (DDIA_Lane_Capability_Control):** This bit selects how lanes are shared between DDI A and DDI E. This field is only used in the DDI A instance of this register. See the DDI A and DDI E lane mapping table in the Introduction section. Restriction : This field must be programmed at system boot based on board configuration and may not be changed afterwards. Restriction : DDI A x2 (DDI A x2 + DDI E x2) selection is not valid on boards that use DDI F.
3:1	0h RW	**DP PORT WIDTH SELECTION (DP_Port_Width_Selection):** This bit selects the number of lanes to be enabled on the DDI link for DisplayPort. This field is ignored for HDMI and DVI which always use all 4 lanes. Restriction : When in DisplayPort mode the value selected here must match the value selected in the DDI Buffer Control register for the DDI attached to this pipe. Restriction : This field must not be changed while the DDI is enabled. DDI E only supports x1 and x2 when DDI_BUF_CTL_A DDIA Lane Capability Control is set to DDIA x2, otherwise DDI E is not supported. DDI A (EDP) supports x1, x2, and x4 when DDI_BUF_CTL_A DDIA Lane Capability Control is set to DDIA x4, otherwise DDI A only supports x1 and x2.
0	0h RO	**INIT DISPLAY DETECTED (Init_Display_Detected):** Strap indicating whether a display was detected on this port during initialization. It signifies the level of the port detect pin at boot. This bit is only informative. It does not prevent this port from being enabled in hardware. This field only indicates the DDIA detection. Detection for other ports is read from SFUSE_STRAP.

32.85 Hot Plug Control (HOTPLUG_CTL)—Offset C4030h

Hot plug detect (HPD) is used for notification of plug, unplug, and other sink events. The short pulse durations are programmed in HPD_PULSE_CNT. The hotplug ISR gives the live states of the HPD pins.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:29	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'Hot Plug Control' does not have a description in the BXML
28	0h RW	**DDI A HPD INPUT ENABLE (DDI_A_HPD_Input_Enable):** This field enables the HPD buffer for digital port A.
27	0h RW	**DDI A HPD INVERT (DDI_A_HPD_Invert):** This field inverts the HPD sense for digital port A.
26	0h RSV	**RESERVED (Reserved_1):** The field 'Reserved' in register 'Hot Plug Control' does not have a description in the BXML
25:24	0h RW/C	**DDI A HPD STATUS (DDI_A_HPD_Status):** This field indicates the hot plug detect status on port A. When HPD input is enabled and either a long or short pulse is detected, one of these bits will set and the hotplug IIR will be set (if unmasked in the IMR). These are sticky bits, cleared by writing 1s to both of them.
23:13	0h RSV	**RESERVED (Reserved_2):** The field 'Reserved' in register 'Hot Plug Control' does not have a description in the BXML
12	0h RW	**DDI C HPD INPUT ENABLE (DDI_C_HPD_Input_Enable):** This field enables the HPD buffer for digital port C.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**DDI C HPD INVERT (DDI_C_HPD_Invert):** This field inverts the HPD sense for digital port C.
10	0h RSV	**RESERVED (Reserved_3):** The field 'Reserved' in register 'Hot Plug Control' does not have a description in the BXML
9:8	0h RW/C	**DDI C HPD STATUS (DDI_C_HPD_Status):** This field indicates the hot plug detect status on port C. When HPD input is enabled and either a long or short pulse is detected, one of these bits will set and the hotplug IIR will be set (if unmasked in the IMR). These are sticky bits, cleared by writing 1s to both of them.
7:5	0h RSV	**RESERVED (Reserved_4):** The field 'Reserved' in register 'Hot Plug Control' does not have a description in the BXML
4	0h RW	**DDI B HPD INPUT ENABLE (DDI_B_HPD_Input_Enable):** This field enables the HPD buffer for digital port B.
3	0h RW	**DDI B HPD INVERT (DDI_B_HPD_Invert):** This field inverts the HPD sense for digital port B.
2	0h RSV	**RESERVED (Reserved_5):** The field 'Reserved' in register 'Hot Plug Control' does not have a description in the BXML
1:0	0h RW/C	**DDI B HPD STATUS (DDI_B_HPD_Status):** This field indicates the hot plug detect status on port B. When HPD input is enabled and either a long or short pulse is detected, one of these bits will set and the hotplug IIR will be set (if unmasked in the IMR). These are sticky bits, cleared by writing 1s to both of them.

32.86 HPD Pulse Count DDI B (HPD_PULSE_CNT_B)—Offset C4034h

There is one instance of this register per DDI. This register is on the chip reset, not the FLR or display debug reset.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 7CEh

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'HPD Pulse Count DDI B' does not have a description in the BXML
16:0	7CEh RW	**SHORTPULSE COUNT (ShortPulse_Count):** These bits define the duration of the pulse defined as a short pulse for DDI HPD. The value is the number of microseconds minus 2.

32.87 HPD Filter Count (HPD_FILTER_CNT)—Offset C4038h

This register is on the chip reset, not the FLR or display debug reset.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 1F2h

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'HPD Filter Count' does not have a description in the BXML
16:0	1F2h RW	**HPD FILTER COUNT (HPD_Filter_Count):** These bits define the duration of the filter for DDI HPD. The value is the number of microseconds minus 2.

32.88 HPD Pulse Count DDI C (HPD_PULSE_CNT_C)—Offset C4044h

There is one instance of this register per DDI. This register is on the chip reset, not the FLR or display debug reset.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 7CEh

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'HPD Pulse Count DDI C' does not have a description in the BXML
16:0	7CEh RW	**SHORTPULSE COUNT (ShortPulse_Count):** These bits define the duration of the pulse defined as a short pulse for DDI HPD. The value is the number of microseconds minus 2.

32.89 HPD Pulse Count DDI A (HPD_PULSE_CNT_A)—Offset C404Ch

There is one instance of this register per DDI. This register is on the chip reset, not the FLR or display debug reset.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 7CEh

Bit Range	Default & Access	Field Name (ID): Description
31:17	0h RSV	**RESERVED (Reserved_0):** The field 'Reserved' in register 'HPD Pulse Count DDI A' does not have a description in the BXML
16:0	7CEh RW	**SHORTPULSE COUNT (ShortPulse_Count):** These bits define the duration of the pulse defined as a short pulse for DDI HPD. The value is the number of microseconds minus 2.

32.90 FENCE0_LSB (FENCE0_LSB)—Offset 100000h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.91 FENCE0_MSB (FENCE0_MSB)—Offset 100004h

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.92 FENCE1_LSB (FENCE1_LSB)—Offset 100008h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.93 FENCE1_MSB (FENCE1_MSB)—Offset 10000Ch

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.94 FENCE2_LSB (FENCE2_LSB)—Offset 100010h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.95 FENCE2_MSB (FENCE2_MSB)—Offset 100014h

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.96 FENCE3_LSB (FENCE3_LSB)—Offset 100018h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.97 FENCE3_MSB (FENCE3_MSB)—Offset 10001Ch

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.98 FENCE4_LSB (FENCE4_LSB)—Offset 100020h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.99 FENCE4_MSB (FENCE4_MSB)—Offset 100024h

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.100 FENCE5_LSB (FENCE5_LSB)—Offset 100028h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.101 FENCE5_MSB (FENCE5_MSB)—Offset 10002Ch

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.102 FENCE6_LSB (FENCE6_LSB)—Offset 100030h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.103 FENCE6_MSB (FENCE6_MSB)—Offset 100034h

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.104 FENCE7_LSB (FENCE7_LSB)—Offset 100038h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.105 FENCE7_MSB (FENCE7_MSB)—Offset 10003Ch

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.106 FENCE8_LSB (FENCE8_LSB)—Offset 100040h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.107 FENCE8_MSB (FENCE8_MSB)—Offset 100044h

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.108 FENCE9_LSB (FENCE9_LSB)—Offset 100048h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.109 FENCE9_MSB (FENCE9_MSB)—Offset 10004Ch

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.110 FENCE10_LSB (FENCE10_LSB)—Offset 100050h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.111 FENCE10_MSB (FENCE10_MSB)—Offset 100054h

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.112 FENCE11_LSB (FENCE11_LSB)—Offset 100058h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.113 FENCE11_MSB (FENCE11_MSB)—Offset 10005Ch

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.114 FENCE12_LSB (FENCE12_LSB)—Offset 100060h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.115 FENCE12_MSB (FENCE12_MSB)—Offset 100064h

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.116 FENCE13_LSB (FENCE13_LSB)—Offset 100068h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.117 FENCE13_MSB (FENCE13_MSB)—Offset 10006Ch

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.118 FENCE14_LSB (FENCE14_LSB)—Offset 100070h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.119 FENCE14_MSB (FENCE14_MSB)—Offset 100074h

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.120 FENCE15_LSB (FENCE15_LSB)—Offset 100078h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.121 FENCE15_MSB (FENCE15_MSB)—Offset 10007Ch

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.122 FENCE16_LSB (FENCE16_LSB)—Offset 100080h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.123 FENCE16_MSB (FENCE16_MSB)—Offset 100084h

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.124 FENCE17_LSB (FENCE17_LSB)—Offset 100088h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.125 FENCE17_MSB (FENCE17_MSB)—Offset 10008Ch

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.126 FENCE18_LSB (FENCE18_LSB)—Offset 100090h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.127 FENCE18_MSB (FENCE18_MSB)—Offset 100094h

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.128 FENCE19_LSB (FENCE19_LSB)—Offset 100098h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.129 FENCE19_MSB (FENCE19_MSB)—Offset 10009Ch

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.130 FENCE20_LSB (FENCE20_LSB)—Offset 1000A0h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.131 FENCE20_MSB (FENCE20_MSB)—Offset 1000A4h

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.132 FENCE21_LSB (FENCE21_LSB)—Offset 1000A8h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.133 FENCE21_MSB (FENCE21_MSB)—Offset 1000ACh

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.134 FENCE22_LSB (FENCE22_LSB)—Offset 1000B0h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.135 FENCE22_MSB (FENCE22_MSB)—Offset 1000B4h

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.136 FENCE23_LSB (FENCE23_LSB)—Offset 1000B8h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.137 FENCE23_MSB (FENCE23_MSB)—Offset 1000BCh

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.138 FENCE24_LSB (FENCE24_LSB)—Offset 1000C0h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.139 FENCE24_MSB (FENCE24_MSB)—Offset 1000C4h

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.140 FENCE25_LSB (FENCE25_LSB)—Offset 1000C8h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.141 FENCE25_MSB (FENCE25_MSB)—Offset 1000CCh

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.142 FENCE26_LSB (FENCE26_LSB)—Offset 1000D0h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.143 FENCE26_MSB (FENCE26_MSB)—Offset 1000D4h

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.144 FENCE27_LSB (FENCE27_LSB)—Offset 1000D8h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.145 FENCE27_MSB (FENCE27_MSB)—Offset 1000DCh

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.146 FENCE28_LSB (FENCE28_LSB)—Offset 1000E0h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.147 FENCE28_MSB (FENCE28_MSB)—Offset 1000E4h

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.148 FENCE29_LSB (FENCE29_LSB)—Offset 1000E8h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.149 FENCE29_MSB (FENCE29_MSB)—Offset 1000ECh

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.150 FENCE30_LSB (FENCE30_LSB)—Offset 1000F0h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.151 FENCE30_MSB (FENCE30_MSB)—Offset 1000F4h

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.152 FENCE31_LSB (FENCE31_LSB)—Offset 1000F8h

Fence Registers LSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCELO (FENCELO):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Lower Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11:2	0h RSV	**RESERVED (RSVD_0):** Reserved
1	0h RW	**TILE (TILE):** This field specifies the spatial ordering of QW within tiles. 0b - Consecutive SWords (32B) sequenced in the X direction 1b - Consecutive OWords (16B) sequenced in the Y direction
0	0h RW/V	**FENCEVAL (FENCEVAL):** This field specifies whether or not this fence register defines a fence region. 0b - FENCE INVALID 1b - FENCE VALID

32.153 FENCE31_MSB (FENCE31_MSB)—Offset 1000FCh

Fence Registers MSBs

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FENCEUP (FENCEUP):** Bits 31:12 of the ending Graphics Address of the fence region. Fence regions must be aligned to a 4KB page. This address represents the last 4KB page of the fence region (Upper Bound is included in the fence region). Graphics Address is the offset within GMADR space.
11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:0	0h RW	**PITCH (Pitch):** This field specifies the width (pitch) of the fence region in multiple of tile widths. For Tile X this field must be programmed to a multiple of 512B (003 is the minimum value) and for Tile Y this field must be programmed to a multiple of 128B (000 is the minimum value). 000h = 128B 001h = 256B 002h = 384B 003h = 512B 004h = 640B 005h = 768B 006h = 896B 007h = 1024B ... 3FFh = 128KB 4FFh = 160KB 5FFh = 192KB 6FFh = 224KB 7FFh = 256KB

32.154 DPFC_CONTROL_SA (DPFC_CTL_SA)—Offset 100100h

This register contains control bits related to Display Frame Buffer Compression Host Invalidation in System Agent.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:30	0h RSV	**RESERVED (RSVD_0):** Reserved
29	0h RW/V	**CPUFNCEN (CPUFNCEN):** 0: Display Buffer is not in a CPU fence. No modifications are allowed from CPU to the Display Buffer. 1: Display Buffer exists in a CPU fence.
28:5	0h RSV	**RESERVED (RSVD_1):** Reserved
4:0	0h RW/V	**CPUFNCNUM (CPUFNCNUM):** This field specifies the CPU visible FENCE number corresponding to the placement of the uncompressed frame buffer.

32.155 DPFC_CPU_FENCE_OFFSET (DPFC_CFO)—Offset 100104h

This register contains control bits related to Display Frame Buffer Compression Host Invalidation in System Agent.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:22	0h RSV	**RESERVED (RSVD_0):** Reserved
21:0	0h RW/V	**YFNCDISP (YFNCDISP):** Y offset from the CPU fence to the Display Buffer base

32.156 TILECTL (TILECTL)—Offset 101000h

Tile control and TLB control.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:3	0h RSV	**RESERVED (RSVD_0):** Reserved
2	0h RSV	**RESERVED (RSVD_1):** Reserved.
1	0h RW/V	**TLBPF (TLBPF):** Store multiple PTE enable. 0: Only one Page Table Entry is stored in the Translation Lookaside Buffer cache. 1: Multiple Page Table Entries (8) are stored in the Translation Lookaside Buffer cache.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW/V	**SWZCTL (SWZCTL):** In order to spread DRAM accesses between multiple channels in the most efficient way, address bits can be used as a channel select. The Driver needs to obtain the need for memory address swizzling via DRAM configuration registers and set the following bits. x0b - No Address Swizzling x1b - Address bit [6] needs to be swizzled for tiled surfaces SWZCTL background (Display has an associated bit) : In order to spread DRAM accesses between multiple channels in the most efficient way, address bits can be used as a channel select. The most common DRAM configuration is 2-channels with 64B interleaving where address bit[6] is used as a channel select. Tiled surface topologies can lead to back-to-back accesses to the same memory channel. In order to minimize these accesses and therefore more efficiently utilize memory bandwidth, we manipulate (or 'swizzle') the tiled address bit[6]. For X-tiled surfaces: A'6 = A6 xor A9 xor A10. Where A represents the tiled address before swizzling and A' represents the address after swizzling.

32.157 GFX_FLSH_CNT (GFX_FLSH_CNT)—Offset 101008h

Used to flush Gunit TLB

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RSV	**RESERVED (RSVD_0):** Reserved
0	0h WO	**GFXFLSHCNTL (GfxFlshCntl):** Access type of this register is WO. A write to this bit flushes the Gfx TLB in GUNIT. The data associated with the write is discarded and a read return all 0s.

32.158 DISPLAY_DEADLINE_CONTROL (DISPLAY_DEADLINE_CONTROL)—Offset 10102Ch

Deadline latency override control

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:11	0h RSV	**RESERVED (RSVD_0):** Reserved
10:1	0h RW/V	**LATENCY_OVERRIDE_OFFSET (Latency_Override_Offset):** When enable, provides a fixed latency offset override for display VC1 requests. For example, value of 0 would sent every VC1 request with the current global time value .. effectively making every VC1 request access 'urgent'.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW/V	**LATENCY_OVERRIDE_ENABLE (Latency_Override_Enable):** 0 (default) : Hardware determined deadlines. 1 : Latency override enabled. Deadline latency offsets are no longer hardware determined. Instead, deadline latency offsets are taken from the 'latency_override_offset' bits

32.159 DG_CLKREQ_POLICY (DG_CLKREQ_POLICY)—Offset 101038h

Various bits that control clock req functionality in DG. This register should be programmed as part of graphics initialization.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW/V	**SPARE2 (SPARE2):** Reserved
23:16	0h RW/V	**CLKREQ_HYST_CNTR (CLKREQ_HYST_CNTR):** Register to store the parameter used to counting IDLE cycles during Hysteris state. Deassert clkreq when DG is idle for the number of cycles this register is programmed to.
15:2	0h RW/V	**SPARE1 (SPARE1):** Reserved
1	0h RW/V	**MEM_UP_OVRD (MEM_UP_OVRD)**
0	0h RW/V	**CLKREQ_OVRD (CLKREQ_OVRD)**

32.160 VDMBDFBARKVM (VDMBDFBARKVM)—Offset 101070h

Allows indirection of KVM traffic for manageability.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 700B0h

Bit Range	Default & Access	Field Name (ID): Description
31:19	0h RSV	**RESERVED (RSVD_0):** Reserved
18:16	7h RW	**BARNUM (BARNUM):** Indicates to which base address register VDM packets should be addressed.

Bit Range	Default & Access	Field Name (ID): Description
15:8	0h RW	**BUSNUM (BUSNUM):** Indicates to which bus number VDM packets should be addressed.
7:3	16h RW	**DEVNUM (DEVNUM):** Indicates to which Device number VDM packets should be addressed.
2:0	0h RW	**FUNNUM (FUNNUM):** Indicates to which Function number VDM packets should be addressed.

32.161 WDBDF (WDBDF)—Offset 101074h

Its the BDF used for wireless display to talk to WNIC. BIOS must program this register with the PCI Bus, Device, and Function.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 100000h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**BUS (BUS):** This field specifies the PCI bus number used for wireless display to talk to WNIC.
23:19	2h RW	**DEVICE (DEVICE):** This field specifies the PCI device number used for wireless display to talk to WNIC.
18:16	0h RW	**FUNCTION (FUNC):** This field specifies the PCI function number used for wireless display to talk to WNIC.
15:0	0h RSV	**RESERVED (RSVD_0):** Reserved

32.162 TOUCHBDF (TOUCHBDF)—Offset 101078h

BIOS must program this register with the PCI Bus, Device, and Function that the Display Engine should use for communication with the PCH Touch Device.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 100000h

Bit Range	Default & Access	Field Name (ID): Description
31:24	0h RW	**BUS (BUS):** This field specifies the PCI bus number of the PCH Touch device.
23:19	2h RW	**DEVICE (DEVICE):** This field specifies the PCI device number of the PCH Touch device.

Bit Range	Default & Access	Field Name (ID): Description
18:16	0h RW	**FUNCTION (FUNC):** This field specifies the PCI function number of the PCH Touch device.
15:0	0h RSV	**RESERVED (RSVD_0):** Reserved

32.163 MTOLUD (MTOLUD)—Offset 108000h

This 32 bit register defines the Top of Low Usable DRAM. TSEG, GTT Graphics memory and Graphics Stolen Memory are within the DRAM space defined. From the top, the Host optionally claims 1 to 64MBs of DRAM for internal graphics if enabled, 1or 2MB of DRAM for GTT Graphics Stolen Memory (if enabled) and 1, 2, or 8 MB of DRAM for TSEG if enabled. Programming Example: C1DRB3 is set to 4GB TSEG is enabled and TSEG size is set to 1MB Internal Graphics is enabled, and Graphics Mode Select is set to 32MB GTT Graphics Stolen Memory Size set to 2MB BIOS knows the OS requires 1G of PCI space. BIOS also knows the range from 0_FEC0_0000h to 0_FFFF_FFFFh is not usable by the system. This 20MB range at the very top of addressable memory space is lost to APIC and LT. According to the above equation, TOLUD is originally calculated to: 4GB = 1_0000_0000h The system memory requirements are: 4GB (max addressable space) - 1GB (pci space) - 35MB (lost memory) = 3GB - 35MB (minimum granularity) = 0_ECB0_0000h Since 0_ECB0_0000h (PCI and other system requirements) is less than 1_0000_0000h, TOLUD should be programmed to ECBh. These bits are Intel TXT lockable.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 100000h

Bit Range	Default & Access	Field Name (ID): Description
31:20	1h RO/V	**TOLUD (TOLUD):** This register contains bits 31 to 20 of an address one byte above the maximum DRAM memory below 4G that is usable by the operating system. Address bits 31 down to 20 programmed to 01h implies a minimum memory size of 1MB. Configuration software must set this value to the smaller of the following 2 choices: maximum amount memory in the system minus ME stolen memory plus one byte or the minimum address allocated for PCI memory. Address bits 19:0 are assumed to be 0_0000h for the purposes of address comparison. The Host interface positively decodes an address towards DRAM if the incoming address is less than the value programmed in this register. The Top of Low Usable DRAM is the lowest address above both Graphics Stolen memory and Tseg. BIOS determines the base of Graphics Stolen Memory by subtracting the Graphics Stolen Memory Size from TOLUD and further decrements by Tseg size to determine base of Tseg. All the Bits in this register are locked in LT mode. This register must be 1MB aligned when reclaim is enabled.
19:1	0h RSV	**RESERVED (RSVD_0):** Reserved
0	0h RO/V	**SPARE (SPARE):** Was lock bit prior to Gen10

32.164 MMIO Mirror of GMCH Graphics Control Register (MGGC)—Offset 108040h

All the bits in this register are Intel TXT lockable.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 500h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RSV	**RESERVED (RSVD_0):** Reserved
15:8	5h RO/V	**GMS (GMS):** This field is used to select the amount of Main Memory that is pre-allocated to support the Internal Graphics device in VGA (non-linear) and Native (linear) modes. It corresponds to DSM (Data Stolen Memory region) region. The BIOS ensures that memory is pre-allocated only when Internal graphics is enabled. Hardware does not clear or set any of these bits automatically based on IGD being disabled/enabled. BIOS Requirement: BIOS must not set this field to 0h if IVD (bit 1 of this register) is 0. BIOS Requirement: Given new sizes allow down to 8MB allocation, BIOS has to ensure there is sufficient space for WOPCM and basic GFX Stolen functions. 00h:0MB 01h:32MB 02h:64MB 03h:96MB 04h:128MB 05h:160MB 06h:192MB 07h:224MB 08h:256MB 09h:288MB 0Ah:320MB 0Bh:352MB 0Ch:384MB 0Dh:416MB 0Eh:448MB 0Fh:480MB 10h:512MB 11h - 1Fh: Reserved 20h:1024MB 21h - 2Fh: Reserved 30h:1536MB 31h - 3Fh: Reserved 40h: 2048MB 41h - EFh: Reserved F0h: 4MB F1h: 8MB F2h: 12MB F3h: 16MB F4h: 20MB F5h: 24MB F6h: 28MB F7h: 32MB F8h: 36MB F9h: 40MB FAh: 44MB FBh: 48MB FCh: 52MB FDh: 56MB FEh: 60MB FFh: Reserved Hardware functionality in case of programming this value to Reserved is not guaranteed.
7:6	0h RO/V	**GGMS (GGMS):** This field is used to select the amount of Main Memory that is pre-allocated to support the Internal Graphics Translation Table. The BIOS ensures that memory is pre-allocated only when Internal graphics is enabled. GSM is assumed to be a contiguous physical DRAM space with DSM, and BIOS needs to allocate a contiguous memory chunk. Hardware will derive the base of GSM from DSM only using the GSM size programmed in the register. Hardware functionality in case of programming this value to Reserved is not guaranteed. 0x0:No Preallocated Memory 0x1:2MB of Preallocated Memory 0x2:4MB of Preallocated Memory 0x3:8MB of Preallocated Memory
5:3	0h RSV	**RESERVED (RSVD_1):** Enables the use of the iGFX enbines for Versatile Acceleration. 1 - iGFX engines are in Versatile Acceleration Mode. Device 2 Class Code is 048000h. 0 - iGFX engines are in iGFX Mode. Device 2 Class Code is 030000h.
2	0h RO/V	**VAMEN (VAMEN):** Reserved
1	0h RO/V	**IVD (IVD):** 0: Enable. Device 2 (IGD) claims VGA memory and IO cycles, the Sub-Class Code within Device 2 Class Code register is 00. 1: Disable. Device 2 (IGD) does not claim VGA cycles (Mem and IO), and the Sub- Class Code field within Device 2 function 0 Class Code register is 80. BIOS Requirement: BIOS must not set this bit to 0 if the GMS field (bits 7:3 of this register) pre-allocates no memory. This bit MUST be set to 1 if Device 2 is disabled either via a fuse or fuse override (CAPID0_A[IGD] = 1) or via a register (DEVEN[3] = 0). This register is locked by Intel TXT lock. 0:Enable 1:Disable
0	0h RO/V	**SPARE (SPARE):** Was lock bit prior to Gen10

32.165 MTOUUD_LSB (MTOUUD_LSB)—Offset 108080h

This 64 bit register defines the Top of Upper Usable DRAM. Configuration software must set this value to TOM minus all ME stolen memory if reclaim is disabled. If reclaim is enabled, this value must be set to reclaim limit + 1byte, 1MB aligned, since reclaim limit is 1MB aligned. Address bits 19:0 are assumed to be 000_0000h for the purposes of address comparison. The Host interface positively decodes an address towards DRAM if the incoming address is less than the value programmed in this register and greater than or equal to 4GB. BIOS Restriction: Minimum value for TOUUD is 4GB. These bits are Intel TXT lockable.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 100000h

Bit Range	Default & Access	Field Name (ID): Description
31:20	1h RO/V	**TOUUD (TOUUD):** This register contains bits 38 to 20 of an address one byte above the maximum DRAM memory above 4G that is usable by the operating system. Configuration software must set this value to TOM minus all ME stolen memory if reclaim is disabled. If reclaim is enabled, this value must be set to reclaim limit 1MB aligned since reclaim limit + 1byte is 1MB aligned. Address bits 19:0 are assumed to be 000_0000h for the purposes of address comparison. The Host interface positively decodes an address towards DRAM if the incoming address is less than the value programmed in this register and greater than 4GB. All the bits in this register are locked in Intel TXT mode.
19:1	0h RSV	**RESERVED (RSVD_0):** Reserved
0	0h RO/V	**SPARE (SPARE):** Was lock bit prior to Gen10

32.166 MTOUUD_MSB (MTOUUD_MSB)—Offset 108084h

This 64 bit register defines the Top of Upper Usable DRAM. Configuration software must set this value to TOM minus all ME stolen memory if reclaim is disabled. If reclaim is enabled, this value must be set to reclaim limit + 1byte, 1MB aligned, since reclaim limit is 1MB aligned. Address bits 19:0 are assumed to be 000_0000h for the purposes of address comparison. The Host interface positively decodes an address towards DRAM if the incoming address is less than the value programmed in this register and greater than or equal to 4GB. BIOS Restriction: Minimum value for TOUUD is 4GB. These bits are Intel TXT lockable.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RO/V	**TOUUD (TOUUD):** This register contains bits63 to 20 of an address one byte above the maximum DRAM memory above 4G that is usable by the operating system. Configuration software must set this value to TOM minus all ME stolen memory if reclaim is disabled. If reclaim is enabled, this value must be set to reclaim limit 1MB aligned since reclaim limit + 1byte is 1MB aligned. Address bits 19:0 are assumed to be 000_0000h for the purposes of address comparison. The Host interface positively decodes an address towards DRAM if the incoming address is less than the value programmed in this register and greater than 4GB. All the bits in this register are locked in Intel TXT mode.

32.167 MBDSM (MBDSM)—Offset 1080C0h

This register contains the base address of graphics data stolen DRAM memory. BIOS determines the base of graphics data stolen memory by subtracting the graphics data stolen memory size (PCI Device 0 offset 52 bits 7:4) from TOLUD (PCI Device 0 offset BC bits 31:20).

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RO/V	**BDSM (BDSM):** This BitField contains bits 31 to 20 of the base address of stolen DRAM memory. BIOS determines the base of graphics stolen memory by subtracting the graphics stolen memory size (PCI Device 0 offset 50 bits 15:8) from TOLUD (PCI Device 0 offset BC bits 31:20).
19:1	0h RSV	**RESERVED (RSVD_0):** Reserved
0	0h RO/V	**SPARE (SPARE):** Was lock bit prior to Gen10

32.168 MBGSM (MBGSM)—Offset 108100h

This register contains the base address of stolen DRAM memory for the GTT. BIOS determines the base of GTT stolen memory by subtracting the GTT graphics stolen memory size (PCI Device 0 offset 52 bits 9:8) from the Graphics Base of Data Stolen Memory (PCI Device 0 offset B0 bits 31:20).

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 100000h

Bit Range	Default & Access	Field Name (ID): Description
31:20	1h RO/V	**BGSM (BGSM):** This register contains the base address of stolen DRAM memory for the GTT. BIOS determines the base of GTT stolen memory by subtracting the GTT graphics stolen memory size (PCI Device 0 offset 50 bits 7:6) from the Graphics Base of Data Stolen Memory (PCI Device 0 offset B0 bits 31:20).
19:1	0h RSV	**RESERVED (RSVD_0):** Reserved
0	0h RO/V	**SPARE (SPARE):** Was lock bit prior to Gen10

32.169 Base of DMA Protected Range (BDPR)—Offset 108140h

This register indicates the Base of DMA Protected Range.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RO/V	**BDPR (BDPR):** 1MB aligned base of DMA Protected Memory Range.
19:0	0h RSV	**RESERVED (RSVD_0):** Reserved

32.170 MPMEN (MPMEN)—Offset 108180h

Register to enable the DMA-protected memory regions setup through the PLMBASE, PLMLIMT, PHMBASE, PHMLIMIT registers. This register is always treated as RO for implementations not supporting protected memory regions (PLMR and PHMR fields reported as Clear in the Capability register). Protected memory regions may be used by software to securely initialize remapping structures in memory. To avoid impact to legacy BIOS usage of memory, software is recommended to not overlap protected memory regions with any reserved memory regions of the platform reported through the Reserved Memory Region Reporting (RMRR) structures.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**EPM (EPM):** This field controls DMA accesses to the protected low-memory and protected high-memory regions. 0: Protected memory regions are disabled. 1: Protected memory regions are enabled. DMA requests accessing protected memory regions are handled as follows: - When DMA remapping is not enabled, all DMA requests accessing protected memory regions are blocked. - When DMA remapping is enabled: -- DMA requests processed as pass-through (Translation Type value of 10b in Context-Entry) and accessing the protected memory regions are blocked. -- DMA requests with translated address (AT=10b) and accessing the protected memory regions are blocked. -- DMA requests that are subject to address remapping, and accessing the protected memory regions may or may not be blocked by hardware. For such requests, software must not depend on hardware protection of the protected memory regions, and instead program the DMA-remapping page-tables to not allow DMA to protected memory regions. Remapping hardware access to the remapping structures are not subject to protected memory region checks. DMA requests blocked due to protected memory region violation are not recorded or reported as remapping faults. Hardware reports the status of the protected memory enable/disable operation through the PRS field in this register. Hardware implementations supporting DMA draining must drain any in-flight translated DMA requests queued within the Root-Complex before indicating the protected memory region as enabled through the PRS field.
30:1	0h RSV	**RESERVED (RSVD_0):** Reserved
0	0h RW	**PRS (PRS):** This field indicates the status of protected memory region(s): 0: Protected memory region(s) disabled. 1: Protected memory region(s) enabled.

32.171 MPLMBASE (MPLMBASE)—Offset 1081C0h

Register to set up the base address of DMA-protected low-memory region below 4GB. This register must be set up before enabling protected memory through PMEN_REG, and must not be updated when protected memory regions are enabled. This register is always treated as RO for implementations not supporting protected low memory region (PLMR field reported as Clear in the Capability register). The alignment of the protected low memory region base depends on the number of reserved bits (N:0) of this register. Software may determine N by writing all 1s to this register, and finding the most significant zero bit position with 0 in the value read back from the register. Bits N:0 of this register is decoded by hardware as all 0s. Software must setup the protected low memory region below 4GB. Software must not modify this register when protected memory regions are enabled (PRS field Set in PMEN_REG).

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW	**PLMB (PLMB):** This register specifies the last host physical address of the DMA-protected low-memory region in system memory.
19:0	0h RSV	**RESERVED (RSVD_0):** Reserved

32.172 MPLMLIMIT (MPLMLIMIT)—Offset 108200h

Register to set up the limit address of DMA-protected low-memory region below 4GB. This register must be set up before enabling protected memory through PMEN_REG, and must not be updated when protected memory regions are enabled. This register is always treated as RO for implementations not supporting protected low memory region (PLMR field reported as Clear in the Capability register). The alignment of the protected low memory region limit depends on the number of reserved bits (N:0) of this register. Software may determine N by writing all 1's to this register, and finding most significant zero bit position with 0 in the value read back from the register. Bits N:0 of the limit register is decoded by hardware as all 1s. The Protected low-memory base and limit registers functions as follows: - Programming the protected low-memory base and limit registers with the same value in bits 31:(N+1) specifies a protected low-memory region of size 2^(N+1) bytes. - Programming the protected low-memory limit register with a value less than the protected low-memory base register disables the protected low-memory region. Software must not modify this register when protected memory regions are enabled (PRS field Set in PMEN_REG).

Access Method

Type: MEM Register (Size: 32 bits)	**Device:** **Function:**

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW	**PLML (PLML):** This register specifies the last host physical address of the DMA-protected low-memory region in system memory.
19:0	0h RSV	**RESERVED (RSVD_0):** Reserved

32.173 MPHMBASE_LSB (MPHMBASE_LSB)—Offset 108240h

Register to set up the base address of DMA-protected high-memory region. This register must be set up before enabling protected memory through PMEN_REG, and must not be updated when protected memory regions are enabled. This register is always treated as RO for implementations not supporting protected high memory region (PHMR field reported as Clear in the Capability register). The alignment of the protected high memory region base depends on the number of reserved bits (N:0) of this register. Software may determine N by writing all 1's to this register, and finding most significant zero bit position below host address width (HAW) in the value read back from the register. Bits N:0 of this register are decoded by hardware as all 0s. Software may setup the protected high memory region either above or below 4GB. Software must not modify this register when protected memory regions are enabled (PRS field Set in PMEN_REG).

Access Method

Type: MEM Register (Size: 32 bits)	**Device:** **Function:**

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW	**PHMB (PHMB):** This register specifies the last host physical address of the DMA-protected high-memory region in system memory. Hardware ignores and does not implement bits 63:HAW, where HAW is the host address width.
19:0	0h RSV	**RESERVED (RSVD_0):** Reserved

32.174 MPHMBASE_MSB (MPHMBASE_MSB)—Offset 108244h

Register to set up the base address of DMA-protected high-memory region. This register must be set up before enabling protected memory through PMEN_REG, and must not be updated when protected memory regions are enabled. This register is always treated as RO for implementations not supporting protected high memory region (PHMR field reported as Clear in the Capability register). The alignment of the protected high memory region base depends on the number of reserved bits (N:0) of this register. Software may determine N by writing all 1's to this register, and finding most significant zero bit position below host address width (HAW) in the value read back from the register. Bits N:0 of this register are decoded by hardware as all 0s. Software may setup the protected high memory region either above or below 4GB. Software must not modify this register when protected memory regions are enabled (PRS field Set in PMEN_REG).

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**PHMB (PHMB):** This register specifies the last host physical address of the DMA-protected high-memory region in system memory. Hardware ignores and does not implement bits 63:HAW, where HAW is the host address width.

32.175 MPHMLIMIT_LSB (MPHMLIMIT_LSB)—Offset 108280h

Register to set up the limit address of DMA-protected high-memory region. This register must be set up before enabling protected memory through PMEN_REG, and must not be updated when protected memory regions are enabled. This register is always treated as RO for implementations not supporting protected high memory region (PHMR field reported as Clear in the Capability register). The alignment of the protected high memory region limit depends on the number of reserved bits (N:0) of this register. Software may determine the value of N by writing all 1's to this register, and finding most significant zero bit position below host address width (HAW) in the value read back from the register. Bits N:0 of the limit register is decoded by hardware as all 1s. The protected high-memory base and limit registers functions as follows. - Programming the protected low-memory base and limit registers with the same value in bits HAW:(N+1) specifies a protected low-memory region of size 2^(N+1) bytes. - Programming the protected high-memory limit register with a value less than the protected high-memory base register disables the protected high-memory region. Software must not modify this register when protected memory regions are enabled (PRS field Set in PMEN_REG).

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:20	0h RW	**PHML (PHML):** This register specifies the last host physical address of the DMA-protected high-memory region in system memory. Hardware ignores and does not implement bits 63:HAW, where HAW is the host address width.
19:0	0h RSV	**RESERVED (RSVD_0):** Reserved

32.176 MPHMLIMIT_MSB (MPHMLIMIT_MSB)—Offset 108284h

Register to set up the limit address of DMA-protected high-memory region. This register must be set up before enabling protected memory through PMEN_REG, and must not be updated when protected memory regions are enabled. This register is always treated as RO for implementations not supporting protected high memory region (PHMR field reported as Clear in the Capability register). The alignment of the protected high memory region limit depends on the number of reserved bits (N:0) of this register. Software may determine the value of N by writing all 1's to this register, and finding most significant zero bit position below host address width (HAW) in the value read back from the register. Bits N:0 of the limit register is decoded by hardware as all 1s. The protected high-memory base and limit registers functions as follows. - Programming the protected low-memory base and limit registers with the same value in bits HAW:(N+1) specifies a protected low-memory region of size 2^(N+1) bytes. - Programming the protected high-memory limit register with a value less than the protected high-memory base register disables the protected high-memory region. Software must not modify this register when protected memory regions are enabled (PRS field Set in PMEN_REG).

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**PHML (PHML):** This register specifies the last host physical address of the DMA-protected high-memory region in system memory. Hardware ignores and does not implement bits 63:HAW, where HAW is the host address width.

32.177 MGCMD (MGCMD)—Offset 108300h

Mirror GT hardware uses for Vtd state.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**TE (TE):** 0: Disable Vtd DMA remapping 1: Enable Vtd DMA remapping
30:26	0h RSV	**RESERVED (RSVD_0):** Reserved
25	0h RO/V	**IRE (IRE):** Used for Debug only. No HW usage model. 0: Disable interrupt-remapping hardware 1: Enable interrupt-remapping hardware
24:0	0h RSV	**RESERVED (RSVD_1):** Reserved

32.178 MEMRR_BASE_LSB (MEMRR_BASE_LSB)—Offset 108340h

The EMRR range is used to protect Xucode memory from unauthorized reads and writes. Any IO access to this range is aborted. This register controls the location of the EMRR range by indicating its starting address. It functions in tandem with the EMRR mask register.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO/V	**RANGE_BASE (RANGE_BASE):** This field corresponds to bits 38:12 of the base address memory range which is allocated to EMRR memory.
11:0	0h RSV	**RESERVED (RSVD_0):** Reserved

32.179 MEMRR_BASE_MSB (MEMRR_BASE_MSB)—Offset 108344h

The EMRR range is used to protect Xucode memory from unauthorized reads and writes. Any IO access to this range is aborted. This register controls the location of the EMRR range by indicating its starting address. It functions in tandem with the EMRR mask register.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RSV	**RESERVED (RSVD_0):** Reserved
6:0	0h RO/V	**RANGE_BASE (RANGE_BASE):** This field corresponds to bits 38:12 of the base address memory range which is allocated to EMRR memory.

32.180 MEMRR_MASK_LSB (MEMRR_MASK_LSB)—Offset 108380h

This register controls the size of the EMRR range by indicating which address bits must match the EMRR base register value.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RO/V	**RANGE_BASE (RANGE_MASK):** This field indicates which address bits must match EMRR base in order to qualify as an EMRR access.
11	0h RO/V	**RANGE_EN (RANGE_EN):** Indicates whether the EMRR range is enabled and valid.
10	0h RO/V	**LOCK (LOCK):** Setting this bit locks all writeable settings in this register, including itself.
9:0	0h RSV	**RESERVED (RSVD_0):** Reserved

32.181 MEMRR_MASK_MSB (MEMRR_MASK_MSB)—Offset 108384h

This register controls the size of the EMRR range by indicating which address bits must match the EMRR base register value.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RSV	**RESERVED (RSVD_0):** Reserved
6:0	0h RO/V	**RANGE_MASK (RANGE_MASK):** This field indicates which address bits must match EMRR base in order to qualify as an EMRR access.

32.182 GTACK (GTACK)—Offset 120004h

This register is written to by GT for various device 2 sequencer flows.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:5	0h RSV	**RESERVED (RSVD_0):** Reserved
4	0h RW/V	**RTPACK (RTPACK):** GT indicates that the set root table pointer flow is complete from its point of view by writing a 1b to this field. Once SW is notified with the appropriate status bit, this bit is cleared by the HW.
3	0h RW/V	**DESCRACK (DESCRACK):** GT indicates that the generic descriptor flow is complete from its point of view by writing a 1b to this field Once SW is notified with the appropriate status bit, this bit is cleared by the HW.
2	0h RW/V	**VTDACK (VTDACK):** GT indicates that the Translation Enable/Disable or IOTLB Invalidation flow is complete from its point of view by writing a 1b to this field. Once SW is notified with the appropriate status bit, this bit is cleared by the HW.
1	0h RW/V	**DPRACK (DPRACK):** GT indicates that DPR Update flow is complete from its point of view by writing a 1b to this field. Once SW is notified with the appropriate status bit, this bit is cleared by the HW.
0	0h RW/V	**PMRACK (PMRACK):** GT indicates that PMR Enable/Disable flow is complete from its point of view by writing a 1b to this field. Once SW is notified with the appropriate status bit, this bit is cleared by the HW.

32.183 EDRAMCAP (EDRAMCAP)—Offset 120010h

Describes the presence and capabilities of the eDRAM cache.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**EDRAMCAP_VALUE (EDRAMCAP_VALUE):** Bit[31] - Reserved Bits[30:26] - EDRAM Frequency, Default Value: 01000b eDRAM frequency: 200MHz* encode. 0x08 = 1600MHz 0x09 = 1800MHz 0x0A = 2000MHz Bits[25:24] - EDRAM_CHANNELS, Default Value: 00b If FUSE_EDRAM_ENABLE=0, this field is NA. 00 = one channel, 01 = two channels. 1x = Reserved. Bits[23:22] - Reserved Bits[21:16] - CURRENT_ETAG_WAYS, Default Value: 11b The total number of ways in each ETAG slice. This field is updated by PCODE after every ETAG shrink or ETAG expand. 0 if 4 ways 1 if 8 ways 2 if 12 ways 3 if 16 ways. Bits[15:10] - Reserved Bits[9:8] - SETS_CONFIGURATION, Default Value: 10b 1K or 2K sets, to support 64MB and 128MB EDRAM size configurations accordingly. 00 = Reserved. 01 if 1K sets 10 if 2K sets 11 reserved. Bits[7:5] - WAYS_CONFIGURATION, Default Value: 011b This field defines the total number of ETAG ways. 000 if 4 ways 001 if 8 ways 010 if 12 ways 011 if 16 ways 1xx reserved. Bits[4:1] - Super Queue Internal Register Count, Bit[0] - FUSE_EDRAM_ENABLE, Default Value: 0b PCODE will update this field based on FUSE_EDRAM_ENABLE.

32.184 MAILBOX0 (MAILBOX0)—Offset 120800h

This register contains bits 31:0 of the generic descriptor, fetched from the invalidate queue for GT.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**DATA (DATA):** This field contains bits 31:0 of the generic descriptor, fetched from the invalidate queue for GT.

32.185 MAILBOX1 (MAILBOX1)—Offset 120804h

This register contains bits 63:32 of the generic descriptor, fetched from the invalidate queue for GT.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**DATA (DATA):** This field contains bits 63:32 of the generic descriptor, fetched from the invalidate queue for GT.

32.186 MAILBOX2 (MAILBOX2)—Offset 120808h

This register contains bits 95:64 of the generic descriptor, fetched from the invalidate queue for GT.

Access Method

Type: MEM Register (Size: 32 bits)	**Device:** **Function:**

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**DATA (DATA):** This field contains bits 95:64 of the generic descriptor, fetched from the invalidate queue for GT.

32.187 MAILBOX3 (MAILBOX3)—Offset 12080Ch

This register contains bits 127:96 of the generic descriptor, fetched from the invalidate queue for GT.

Access Method

Type: MEM Register (Size: 32 bits)	**Device:** **Function:**

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW/V	**DATA (DATA):** This field contains bits 127:96 of the generic descriptor, fetched from the invalidate queue for GT.

32.188 FLT_RPT0 (FLT_RPT0)—Offset 124810h

GT uses this register to post VT-d faults

Access Method

Type: MEM Register (Size: 32 bits)	**Device:** **Function:**

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**FI (FI):** Fault Info.
11:0	0h RSV	**RESERVED (RSVD_0):** The field 'RESERVED' in register 'FLT_RPT0' does not have a description in the BXML

32.189 FLT_RPT1 (FLT_RPT1)—Offset 124814h

GT uses this register to post VT-d faults

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**FI (FI):** Fault Info.

32.190 FLT_RPT2 (FLT_RPT2)—Offset 124818h

GT uses this register to post VT-d faults

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 10h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**PP (PP):** PASID Present.
30	0h RW	**EXE (EXE):** Execute Permission Requested.
29	0h RW	**PRIV (PRIV):** Privilege Mode Requested .
28:16	0h RSV	**RESERVED (RSVD_0):** Reserved
15:0	10h RO	**SOURCE ID (SID):** Source ID.

32.191 FLT_RPT3 (FLT_RPT3)—Offset 12481Ch

GT uses this register to post VT-d faults

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**F (F):** Fault
30	0h RW	**T (T):** Type

Bit Range	Default & Access	Field Name (ID): Description
29:28	0h RW	**AT (AT):** Address Type
27:8	0h RW	**PN (PN):** PASID Number
7:0	0h RW	**FR (FR):** Fault Reason

32.192 PPRO (PPRO)—Offset 124820h

GT uses this register to post Page Request Queue overflow faults

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RSV	**RESERVED (RSVD_0):** Reserved
0	0h WO	**POST PAGE REQUEST OVERFLOW FAULT (PPRO):** Post Page Request overflow fault

32.193 PPPR (PPPR)—Offset 124824h

GT uses this register to post pending page requests to software, such as x86 page faults. A write to this register triggers an MSI per the registers PRESTS, PRECTL, PREDATA, PREADR, and PREUADR.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RSV	**RESERVED (RSVD_0):** Reserved
0	0h WO	**POST PENDING PAGE REQUEST (PPPR):** Post Pending Page Request

32.194 RTADDR_LSB (RTADDR_LSB)—Offset 124830h

Register providing the base address of root-entry table.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**RTA (RTA):** This register points to base of page aligned, 4KB-sized root-entry table in system memory. Hardware ignores and not implements bits 63:HAW, where HAW is the host address width. Software specifies the base address of the root-entry table through this register, and programs it in hardware through the SRTP field in the Global Command register. Reads of this register returns value that was last programmed to it.
11	0h RW	**RTT (RTT):** This field specifies the type of root-table referenced by the Root Table Address (RTA) field; 0: Root Table 1: Extended Root Table
10:0	0h RSV	**RESERVED (RSVD_0):** Reserved

32.195 RTADDR_MSB (RTADDR_MSB)—Offset 124834h

Register providing the base address of root-entry table.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:7	0h RSV	**RESERVED (RSVD_0):** Reserved
6:0	0h RW	**RTA (RTA):** This register points to base of page aligned, 4KB-sized root-entry table in system memory. Hardware ignores and not implements bits 63:HAW, where HAW is the host address width. Software specifies the base address of the root-entry table through this register, and programs it in hardware through the SRTP field in the Global Command register. Reads of this register returns value that was last programmed to it.

32.196 GT Scratchpad 0 (GTSP0)—Offset 130040h

The register 'GT Scratchpad 0' does not have a description in the BXML source

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GT SCRATCH PAD (GT_scratch_pad):** The field 'GT scratch pad' in register 'GT Scratchpad 0' does not have a description in the BXML

32.197 GT Scratchpad 1 (GTSP1)—Offset 130044h

The register 'GT Scratchpad 1' does not have a description in the BXML source

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**GT SCRATCH PAD (GT_scratch_pad):** The field 'GT scratch pad' in register 'GT Scratchpad 1' does not have a description in the BXML
15:0	0h RW	**MULTIPLE FORCE WAKE (Multiple_Force_Wake):** GT programs this field with the multiple force wake status. Software reads this field to find the status. Refer to MULTIFORCEWAKE 0xA188 register description for the usage.

32.198 GT Scratchpad 2 (GTSP2)—Offset 130048h

The register 'GT Scratchpad 2' does not have a description in the BXML source

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GT SCRATCH PAD (GT_scratch_pad):** The field 'GT scratch pad' in register 'GT Scratchpad 2' does not have a description in the BXML

32.199 GT Scratchpad 3 (GTSP3)—Offset 13004Ch

The register 'GT Scratchpad 3' does not have a description in the BXML source

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GT SCRATCH PAD (GT_scratch_pad):** The field 'GT scratch pad' in register 'GT Scratchpad 3' does not have a description in the BXML

32.200 GT Scratchpad 4 (GTSP4)—Offset 130050h

The register 'GT Scratchpad 4' does not have a description in the BXML source

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GT SCRATCH PAD (GT_scratch_pad):** The field 'GT scratch pad' in register 'GT Scratchpad 4' does not have a description in the BXML

32.201 GT Scratchpad 5 (GTSP5)—Offset 130054h

The register 'GT Scratchpad 5' does not have a description in the BXML source

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GT SCRATCH PAD (GT_scratch_pad):** The field 'GT scratch pad' in register 'GT Scratchpad 5' does not have a description in the BXML

32.202 GT Scratchpad 6 (GTSP6)—Offset 130058h

The register 'GT Scratchpad 6' does not have a description in the BXML source

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:0	0h RW	**GT SCRATCH PAD (GT_scratch_pad):** The field 'GT scratch pad' in register 'GT Scratchpad 6' does not have a description in the BXML

32.203 GT Scratchpad 7 (GTSP7)—Offset 13005Ch

The register 'GT Scratchpad 7' does not have a description in the BXML source

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:1	0h RW	**GT SCRATCH PAD (GT_scratch_pad):** The field 'GT scratch pad' in register 'GT Scratchpad 7' does not have a description in the BXML
0	0h RO	Reserved.

32.204 Version Register (VER_REG_0_0_0_VTDBAR)—Offset 0h

Register to report the architecture version supported. Backward compatibility for the architecture is maintained with new revision numbers, allowing software to load remapping hardware drivers written for prior architecture versions.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 10h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7:4	1h RO	**Major Version Number (MAJOR):** Indicates supported architecture version.
3:0	0h RO	**Minor Version Number (MINOR):** Indicates supported architecture minor version.

32.205 Capability Register (CAP_REG_0_0_0_VTDBAR)—Offset 8h

Register to report general remapping hardware capabilities. Note: These values can change based on defeature bits.

Access Method

Type: MEM Register
(Size: 64 bits)

Device:
Function:

Default: 1C0000C40660462h

Bit Range	Default & Access	Field Name (ID): Description
63:57	0h RO	Reserved.

Bit Range	Default & Access	Field Name (ID): Description
56	1h RO	**First Level 64-KByte Page SupportP (FL1GP):** A value of 1 in this field indicates 1-GByte page size is supported for first-level translation. Note: These values can change based on defeature bits.
55	1h RO	**Read Draining (DRD):** • 0 = Hardware does not support draining of DMA read requests. • 1 = Hardware supports draining of DMA read requests. Note: These values can change based on defeature bits.
54	1h RO	**Write Draining (DWD):** • 0 = Hardware does not support draining of DMA write requests. • 1 = Hardware supports draining of DMA write requests. Note: These values can change based on defeature bits.
53:48	0h RO	**Maximum Address Mask Value (MAMV):** The value in this field indicates the maximum supported value for the Address Mask (AM) field in the Invalidation Address register (IVA_REG) and IOTLB Invalidation Descriptor (iotlb_inv_dsc) used for invalidations of second-level translation. This field is valid only when the PSI field in Capability register is reported as Set. Note: These values can change based on defeature bits.
47:40	0h RO	**Number of Fault-Recording Registers (NFR):** Number of fault recording registers is computed as N+1, where N is the value reported in this field. Implementations must support at least one fault recording register (NFR = 0) for each remapping hardware unit in the platform. The maximum number of fault recording registers per remapping hardware unit is 256. Note: These values can change based on defeature bits.
39	0h RO	**Page Selective Invalidation (PSI):** • 0 = Hardware supports only domain and global invalidates for IOTLB.{*}1 = Hardware supports page selective, domain and global invalidates for IOTLB. Hardware implementations reporting this field as set are recommended to support a Maximum Address Mask Value (MAMV) value of at least 9 (or 18 if supporting 1GB pages with second level translation). Note: These values can change based on defeature bits.
38	0h RO	Reserved.
37:34	3h RO	**Second Level Large Page Support (SLLPS):** This field indicates the super page sizes supported by hardware. A value of 1 in any of these bits indicates the corresponding super-page size is supported. The super-page sizes corresponding to various bit positions within this field are: • 0 = 21-bit offset to page frame (2MB) • 1 = 30-bit offset to page frame (1GB) • 2 = 39-bit offset to page frame (512GB) • 3 = 48-bit offset to page frame (1TB) Hardware implementations supporting a specific super-page size must support all smaller super-page sizes, i.e. only valid values for this field are 0000b, 0001b, 0011b, 0111b, 1111b. Note: These values can change based on defeature bits.
33:24	40h RO	**Fault-Recording Register Offset (FRO):** This field specifies the location to the first fault recording register relative to the register base address of this remapping hardware unit. If the register base address is X, and the value reported in this field is Y, the address for the first fault recording register is calculated as X+(16*Y). Note: These values can change based on defeature bits.
23	0h RO	Reserved.
22	1h RO	**Zero Length Read (ZLR):** • 0 = Indicates the remapping hardware unit blocks (and treats as fault) zero length DMA read requests to write-only pages. • 1 = Indicates the remapping hardware unit supports zero length DMA read requests to write-only pages. DMA remapping hardware implementations are recommended to report ZLR field as Set. Note: These values can change based on defeature bits.

Bit Range	Default & Access	Field Name (ID): Description
21:16	26h RO	**Maximum Guest Address Width (MGAW):** This field indicates the maximum DMA virtual addressability supported by remapping hardware. The Maximum Guest Address Width (MGAW) is computed as (N+1), where N is the value reported in this field. For example, a hardware implementation supporting 48-bit MGAW reports a value of 47 (101111b) in this field. If the value in this field is X, untranslated and translated DMA requests to addresses above 2(x+1)-1 are always blocked by hardware. Translations requests to address above 2(x+1)-1 from allowed devices return a null Translation Completion Data Entry with R=W=0. Guest addressability for a given DMA request is limited to the minimum of the value reported through this field and the adjusted guest address width of the corresponding page-table structure. (Adjusted guest address widths supported by hardware are reported through the SAGAW field). Implementations are recommended to support MGAW at least equal to the physical addressability (host address width) of the platform. Note: These values can change based on defeature bits.
15:13	0h RO	Reserved.
12:8	4h RO	**Supported Adjusted Guest Address Widths (SAGAW):** This 5-bit field indicates the supported adjusted guest address widths (which in turn represents the levels of page-table walks for the 4KB base page size) supported by the hardware implementation. A value of 1 in any of these bits indicates the corresponding adjusted guest address width is supported. The adjusted guest address widths corresponding to various bit positions within this field are: • 0 = 30-bit AGAW (2-level page table) • 1 = 39-bit AGAW (3-level page table) • 2 = 48-bit AGAW (4-level page table) • 3 = 57-bit AGAW (5-level page table) • 4 = 64-bit AGAW (6-level page table) Software must ensure that the adjusted guest address width used to setup the page tables is one of the supported guest address widths reported in this field. Note: These values can change based on defeature bits.
7	0h RO	**Caching Mode (CM):** • 0 = Not-present and erroneous entries are not cached in any of the renmapping caches. Invalidations are not required for modifications to individual not present or invalid entries. However, any modifications that result in decreasing the effective permissions or partial permission increases require invalidations for them to be effective. • 1 = Not-present and erroneous mappings may be cached in the remapping caches. Any software updates to the remapping structures (including updates to not-present or erroneous entries) require explicit invalidation. Hardware implementations of this architecture must support a value of 0 in this field. Note: These values can change based on defeature bits.
6	1h RO	**Protected High-Memory Region (PHMR):** • 0 = Indicates protected high-memory region is not supported. • 1 = Indicates protected high-memory region is supported. Note: These values can change based on defeature bits.
5	1h RO	**Protected Low-Memory Region (PLMR):** • 0 = Indicates protected low-memory region is not supported. • 1 = Indicates protected low-memory region is supported. Note: These values can change based on defeature bits.
4	0h RO	**Required Write-Buffer Flushing (RWBF):** • 0 = Indicates no write-buffer flushing is needed to ensure changes to memory-resident structures are visible to hardware. • 1 = Indicates software must explicitly flush the write buffers to ensure updates made to memory-resident remapping structures are visible to hardware. Note: These values can change based on defeature bits.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO	**Advanced Fault Logging (AFL):** • 0 = Indicates advanced fault logging is not supported. Only primary fault logging is supported. • 1 = Indicates advanced fault logging is supported. Note: These values can change based on defeature bits.
2:0	2h RO	**Number of Domains Supported (ND):** • 000b = Hardware supports 4-bit domain-ids with support for up to 16 domains. • 001b = Hardware supports 6-bit domain-ids with support for up to 64 domains. • 010b = Hardware supports 8-bit domain-ids with support for up to 256 domains. • 011b = Hardware supports 10-bit domain-ids with support for up to 1024 domains. • 100b = Hardware supports 12-bit domain-ids with support for up to 4K domains. • 100b = Hardware supports 14-bit domain-ids with support for up to 16K domains. • 110b = Hardware supports 16-bit domain-ids with support for up to 64K domains. • 111b = Reserved. Note: These values can change based on defeature bits.

32.206 Extended Capability Register (ECAP_REG_0_0_0_VTDBAR)—Offset 10h

Register to report remapping hardware extended capabilities. Note: These values can change based on defeature bits.

Access Method

Type: MEM Register
(Size: 64 bits)

Device:
Function:

Default: 19E2FF0505Eh

Bit Range	Default & Access	Field Name (ID): Description
63:41	0h RO	Reserved.
40	1h RO	**Process Address Space ID Support (PASID):** • 0 = Hardware does not support requests tagged with Process Address Space IDs. • 1 = Hardware supports requests tagged with Process Address Space IDs. Note: These values can change based on defeature bits.
39:35	13h RO	**PASID Size Supported (PSS):** This field reports the PASID size supported by the remapping hardware for requests-with-PASID. A value of N in this field indicates hardware supports PASID field of N+1 bits (For example, value of 7 in this field, indicates 8-bit PASIDs are supported). Requests-with-PASID with PASID value beyond the limit specified by this field are treated as error by the remapping hardware. This field is valid only when PASID field is reported as Set. Note: These values can change based on defeature bits.
34	1h RO	**Extended Accessed Flag Support (EAFS):** • 0 = Hardware does not support the extended-accessed (EA) bit in first-level paging-structure entries. • 1 = Hardware supports the extended accessed (EA) bit in first-level paging-structure entries. This field is valid only when PASID field is reported as Set. Note: These values can change based on defeature bits.

Bit Range	Default & Access	Field Name (ID): Description
33	1h RO	**No Write Flag Support (NWFS):** • 0 = Hardware ignores the No Write (NW) flag in Device-TLB translationrequests, and behaves as if NW is always 0. • 1 = Hardware supports the No Write (NW) flag in Device-TLB translationrequests. This field is valid only when Device-TLB support (DT) field is reported as Set. Note: These values can change based on defeature bits.
32	0h RO	**PASID-Only Translations (POT):** • 0 = Hardware does not support PASID-only Translation Type in extended-context-entries. • 1 = Hardware supports PASID-only Translation Type in extended-context-entries. This field is valid only when PASID field is reported as Set. Note: These values can change based on defeature bits.
31	0h RO	**Supervisor Request Support (SRS):** • 0 = H/W does not support requests-with-PASID seeking supervisor privilege. • 1 = H/W supports requests-with-PASID seeking supervisor privilege. The field is valid only when PASID field is reported as Set. Note: These values can change based on defeature bits.
30	0h RO	**Execute Request Support (ERS):** • 0 = H/W does not support requests-with-PASID seeking execute permission. • 1 = H/W supports requests-with-PASID seeking execute permission. This field is valid only when PASID field is reported as Set. Note: These values can change based on defeature bits.
29	1h RO	**Page Request Support (PRS):** • 0 = Hardware does not support Page Requests. • 1 = Hardware supports Page Requests This field is valid only when Device-TLB (DT) field is reported as Set. Note: These values can change based on defeature bits.
28	0h RO	**Ignore (IGN):** Ignore this field
27	1h RO	**Deferred Invalidate Support (DIS):** • 0 = Hardware does not support deferred invalidations of IOTLB and Device-TLB. • 1 = Hardware supports deferred invalidations of IOTLB and Device-TLB. This field is valid only when PASID field is reported as Set. Note: These values can change based on defeature bits.
26	1h RO	**Nested Translation Support (NEST):** • 0 = Hardware does not support nested translations. • 1 = Hardware supports nested translations. This field is valid only when PASID field is reported as Set. Note: These values can change based on defeature bits.
25	1h RO	**Memory Type Support (MTS):** • 0 = Hardware does not support Memory Type in first-level translation and Extended Memory type in second-level translation. • 1 = Hardware supports Memory Type in first-level translation and Extended Memory type in second-level translation. This field is valid only when PASID and ECS fields are reported as Set. Remapping hardware units with, one or more devices that operate in processor coherency domain, under its scope must report this field as Set. Note: These values can change based on defeature bits.
24	1h RO	**Extended Context Support (ECS):** • 0 = Hardware does not support extended-root-entries and extended-context-entries. • 1 = Hardware supports extended-root-entries and extended-context-entries. Implementations reporting PASID or PRS fields as Set, must report this field as Set. Note: These values can change based on defeature bits.

Bit Range	Default & Access	Field Name (ID): Description
23:20	Fh RO	**Maximum Handle Mask Value (MHMV):** The value in this field indicates the maximum supported value for the Handle Mask (HM) field in the interrupt entry cache invalidation descriptor (iec_inv_dsc). This field is valid only when the IR field in Extended Capability register is reported as Set. Note: These values can change based on defeature bits.
19:18	0h RO	Reserved.
17:8	50h RO	**IOTLB Register Offset (IRO):** This field specifies the offset to the IOTLB registers relative to the register base address of this remapping hardware unit. If the register base address is X, and the value reported in this field is Y, the address for the first IOTLB invalidation register is calculated as X+(16*Y). Note: These values can change based on defeature bits.
7	0h RO	**Snoop Control (SC):** • 0 = Hardware does not support 1-setting of the SNP field in the page-table entries. • 1 = Hardware supports the 1-setting of the SNP field in the page-table entries. Note: These values can change based on defeature bits.
6	1h RO	**Pass Through (PT):** • 0 = Hardware does not support pass-through translation type in context entries and extended-context-entries. • 1 = Hardware supports pass-through translation type in context entries and extended-context-entries. Pass-through translation is specified through Translation-Type (T) field value of 10b in context-entries, or T field value of 010b in extended-context-entries. Hardware implementations supporting PASID must report a value of 1b in this field. Note: These values can change based on defeature bits.
5	0h RO	Reserved.
4	1h RO	**Extended Interrupt Mode (EIM):** • 0 = On Intel64 platforms, hardware supports only 8-bit APIC-IDs (xAPIC mode). • 1 = On Intel64 platforms, hardware supports 32-bit APIC-IDs (x2APIC mode). This field is valid only on Intel64 platforms reporting Interrupt Remapping support (IR field Set). Note: These values can change based on defeature bits.
3	1h RO	**Interrupt Remapping Suport (IR):** • 0 = Hardware does not support interrupt remapping. • 1 = Hardware supports interrupt remapping. Implementations reporting this field as Set must also support Queued Invalidation (QI). Note: These values can change based on defeature bits.
2	1h RO	**Device-TLB Support (DT):** • 0 = Hardware does not support device-IOTLBs. • 1 = Hardware supports Device-IOTLBs. Implementations reporting this field as Set must also support Queued Invalidation (QI). Hardware implementations supporting I/O Page Requests (PRS field Set in Extended Capability register) must report a value of 1b in this field. Note: These values can change based on defeature bits.
1	1h RO	**Queued Invalidation Support (QI):** • 0 = Hardware does not support queued invalidations. • 1 = Hardware supports queued invalidations. Note: These values can change based on defeature bits.
0	0h RO	**Page-Walk Coherency (C):** This field indicates if hardware access to the root, context, extended-context and interrupt-remap tables, and second-level paging structures for requests-without-PASID, are coherent (snooped) or not. • 0 = Indicates hardware accesses to remapping structures are non-coherent. • 1 = Indicates hardware accesses to remapping structures are coherent. Hardware access to advanced fault log, invalidation queue, invalidation semaphore, page-request queue, PASID-table, PASID-state table, and first-level page-tables are always coherent. Note: These values can change based on defeature bits.

32.207 Global Command Register (GCMD_REG_0_0_0_VTDBAR)—Offset 18h

Register to control remapping hardware. If multiple control fields in this register need to be modified, software must serialize the modifications through multiple writes to this register.

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Translation Enable (TE):** Software writes to this field to request hardware to enable/disable DMA-remapping: • 0 = Disable DMA remapping. • 1 = Enable DMA remapping. Hardware reports the status of the translation enable operation through the TES field in the Global Status register. There may be active DMA requests in the platform when software updates this field. Hardware must enable or disable remapping logic only at determinstic transaction boundaries, so that any in-flight transaction is either subject to remapping or not at all. Hardware implementations supporting DMA draining must drain any in-flight DMA read/write requests queued within the Root-Complex before completing the translation enable command and reflecting the status of the command through the TES field in the Global Status register. The value returned on a read of this field is undefined. Note: This field behaves like a WO field except for requests with certain SAIs: Power Management, DFX Red2, and DFX Red4.
30	0h WO	**Set Root Table Pointer (SRTP):** Software sets this field to set/update the root-entry table pointer used by hardware. The root-entry table pointer is specified through the Root-entry Table Address (RTA_REG) register. Hardware reports the status of the Set Root Table Pointer operation through the RTPS field in the Global Status register. The Set Root Table Pointer operation must be performed before enabling or re-enabling (after disabling) DMA remapping through the TE field. .After a Set Root Table Pointer operation, software must globally invalidate the context cache and then globally invalidate of IOTLB. This is required to ensure hardware uses only the remapping structures referenced by the new root table pointer, and not stale cached entries. While DMA remapping hardware is active, software may update the root table pointer through this field. However, to ensure valid in-flight DMA requests are deterministically remapped, software must ensure that the structures referenced by the new root table pointer are programmed to provide the same remapping results as the structures referenced by the previous root-table pointer. Clearing this bit has no effect. The value returned on read of this field is undefined. Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
29	0h RO	**Set Fault Log (SFL):** This field is valid only for implementations supporting advanced fault logging. Software sets this field to request hardware to set/update the fault-log pointer used by hardware. The fault-log pointer is specified through Advanced Fault Log register. Hardware reports the status of the Set Fault Log operation through the FLS field in the Global Status register. The fault log pointer must be set before enabling advanced fault logging (through EAFL field). Once advanced fault logging is enabled, the fault log pointer may be updated through this field while DMA remapping is active. Clearing this bit has no effect. The value returned on read of this field is undefined.

Bit Range	Default & Access	Field Name (ID): Description
28	0h RO	**Enable Advanced Fault Logging (EAFL):** This field is valid only for implementations supporting advanced fault logging. Software writes to this field to request hardware to enable or disable advanced fault logging: • 0 = Disable advanced fault logging. In this case, translation faults are reported through the Fault Recording registers. • 1 = Enable use of memory-resident fault log. When enabled, translation faults are recorded in the memory-resident log. The fault log pointer must be set in hardware (through the SFL field) before enabling advaned fault logging. Hardware reports the status of the advaned fault logging enable operation through the AFLS field in the Global Status register. The value returned on read of this field is undefined.
27	0h RO	**Write Buffer Flush (WBF):** This bit is valid only for implementations requiring write buffer flushing. Software sets this field to request that hardware flush the Root-Complex internal write buffers. This is done to ensure any updates to the memory-resident remapping structures are not held in any internal write posting buffers. Hardware reports the status of the write buffer flushing operation through the WBFS field in the Global Status register. Clearing this bit has no effect. The value returned on a read of this field is undefined.
26	0h RW	**Queued Invalidation Enable (QIE):** This field is valid only for implementations supporting queued invalidations. Software writes to this field to enable or disable queued invalidations. • 0 = Disable queued invalidations. • 1 = Enable use of queued invalidations. Hardware reports the status of queued invalidation enable operation through QIES field in the Global Status register. The value returned on a read of this field is undefined. Note: This field behaves like a WO field except for requests with certain SAIs: Power Management, DFX Red2, and DFX Red4.
25	0h RW	**Interrupt Remapping Enable (IRE):** This field is valid only for implementations supporting interrupt remapping. • 0 = Disable interrupt-remapping hardware. • 1 = Enable interrupt-remapping hardware. Hardware reports the status of the interrupt remapping enable operation through the IRES field in the Global Status register. There may be active interrupt requests in the platform when software updates this field. Hardware must enable or disable interrupt-remapping logic only at deterministic transaction boundaries, so that any in-flight interrupts are either subject to remapping or not at all. Hardware implementations must drain any in-flight interrupts requests queued in the Root-Complex before completing the interrupt-remapping enable command and reflecting the status of the command through the IRES field in the Global Status register. The value returned on a read of this field is undefined. Note: This field behaves like a WO field except for requests with certain SAIs: Power Management, DFX Red2, and DFX Red4.
24	0h WO	**Set Interrupt Remap Table Pointer (SIRTP):** This field is valid only for implementations supporting interrupt-remapping. Software sets this field to set/update the interrupt remapping table pointer used by hardware. The interrupt remapping table pointer is specified through the Interrupt Remapping Table Address (IRTA_REG) register. Hardware reports the status of the Set Interrupt Remap Table Pointer operation through the IRTPS field in the Global Status register. The Set Interrupt Remap Table Pointer operation must be performed before enabling or re-enabling (after disabling) interrupt-remapping hardware through the IRE field. After a Set Interrupt Remap Table Pointer operation, software must globally invalidate the interrupt entry cache. This is required to ensure hardware uses only the interrupt-remapping entries referenced by the new interrupt remap table pointer, and not any stale cached entries. While interrupt remapping is active, software may update the interrupt remapping table pointer through this field. However, to ensure valid in-flight interrupt requests are deterministically remapped, software must ensure that the structures referenced by the new interrupt remap table pointer are programmed to provide the same remapping results as the structures referenced by the previous interrupt remap table pointer. Clearing this bit has no effect. The value returned on a read of this field is undefined. Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**Compatibility Format Interrupt (CFI):** This field is valid only for Intel64 implementations supporting interrupt-remapping. Software writes to this field to enable or disable Compatibility Format interrupts on Intel64 platforms. The value in this field is effective only when interrupt-remapping is enabled and Extended Interrupt Mode (x2APIC mode) is not enabled. • 0 = Block Compatibility format interrupts. • 1 = Process Compatibility format interrupts as pass-through (bypass interrupt remapping). Hardware reports the status of updating this field through the CFIS field in the Global Status register. The value returned on a read of this field is undefined. Note: This field behaves like a WO field except for requests with certain SAIs: Power Management, DFX Red2, and DFX Red4.
22:0	0h RO	Reserved.

32.208 Global Status Register (GSTS_REG_0_0_0_VTDBAR)—Offset 1Ch

Register to report general remapping hardware status.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Translation Enable Status (TES):** This field indicates the status of DMA-remapping hardware. • 0 = DMA-remapping hardware is not enabled. • 1 = DMA-remapping hardware is enabled Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
30	0h RO/V	**Root Table Pointer Status (RTPS):** This field indicates the status of the root- table pointer in hardware. This field is cleared by hardware when software sets the SRTP field in the Global Command register. This field is set by hardware when hardware completes the Set Root Table Pointer operation using the value provided in the Root-Entry Table Address register. Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
29	0h RO	**Fault Log Status (FLS):** This field: • Is cleared by hardware when software Sets the SFL field in the Global Command register. • Is Set by hardware whn hardware completes the Set Fault Log Pointer operation using the value provided in the Advanced Fault Log register.
28	0h RO	**Adavanced Fault Logging Status (AFLS):** This field is valid only for implementations supporting advanced fault logging. It indicates the advanced fault logging status: • 0 = Advanced Fault Logging is not enabled. • 1 = Advanced Fault Logging is enabled.

Bit Range	Default & Access	Field Name (ID): Description
27	0h RO	**Write Buffer Flush Status (WBFS):** This field is valid only for implementations requiring write buffer flushing. This field indicates the status of the write buffer flush command. It is: • Set by hardware when software sets the WBF field in the Global Command register. • Cleared by hardware when hardware completes the write buffer flushing operation.
26	0h RO/V	**Queued Invalidation Enable Status (QIES):** This field indicates queued invalidation enable status. • 0 = queued invalidation is not enabled. • 1 = queued invalidation is enabled Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
25	0h RO/V	**Interrupt Remapping Enable Status (IRES):** This field indicates the status of Interrupt-remapping hardware. • 0 = Interrupt-remapping hardware is not enabled. • 1 = Interrupt-remapping hardware is enabled Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
24	0h RO/V	**Interrupt Remapping Pointer Status (IRTPS):** This field indicates the status of the interrupt remapping table pointer in hardware. This field is cleared by hardware when software sets the SIRTP field in the Global Command register. This field is Set by hardware when hardware completes the set interrupt remap table pointer operation using the value provided in the Interrupt Remapping Table Address register. Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
23	0h RO/V	**Compatibility Format Interrupt Status (CFIS):** This field indicates the status of Compatibility format interrupts on Intel64 implementations supporting interrupt-remapping. The value reported in this field is applicable only when interrupt-remapping is enabled and Extended Interrupt Mode (x2APIC mode) is not enabled. • 0 = Compatibility format interrupts are blocked. • 1 = Compatibility format interrupts are processed as pass-through (bypassing interrupt remapping). Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
22:0	0h RO	Reserved.

32.209 Root Table Address Register (RTADDR_REG_0_0_0_VTDBAR)—Offset 20h

Register providing the base address of root-entry table.

Access Method

Type: MEM Register
(Size: 64 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:12	0h RW	**Root Table Address (RTA):** This register points to base of page aligned, 4KB-sized root-entry table in system memory. Hardware ignores and not implements bits 63:HAW, where HAW is the host address width. Software specifies the base address of the root-entry table through this register, and programs it in hardware through the SRTP field in the Global Command register. Reads of this register returns value that was last programmed to it.

Bit Range	Default & Access	Field Name (ID): Description
11	0h RW	**Root Table Type (RTT):** This field specifies the type of root-table referenced by the Root Table Address (RTA) field: • 0 = Root Table. • 1 = Extended Root Table
10:0	0h RO	Reserved.

32.210 Context Command Register (CCMD_REG_0_0_0_VTDBAR)—Offset 28h

Register to manage context cache. The act of writing the uppermost byte of the CCMD_REG with the ICC field Set causes the hardware to perform the context-cache invalidation.

Access Method

Type: MEM Register
(Size: 64 bits)

Device:
Function:

Default: 8000000000000000h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW/V	**Invalidate Context Cache (ICC):** Software requests invalidation of context-cache by setting this field. Software must also set the requested invalidation granularity by programming the CIRG field. Software must read back and check the ICC field is Clear to confirm the invalidation is complete. Software must not update this register when this field is set. Hardware clears the ICC field to indicate the invalidation request is complete. Hardware also indicates the granularity at which the invalidation operation was performed through the CAIG field. Software must submit a context-cache invalidation request through this field only when there are no invalidation requests pending at this remapping hardware unit. Since information from the context-cache may be used by hardware to tag IOTLB entries, software must perform domain-selective (or global) invalidation of IOTLB after the context cache invalidation has completed. Hardware implementations reporting write-buffer flushing requirement (RWBF=1 in Capability register) must implicitly perform a write buffer flush before invalidating the context cache.
62:61	0h RW	**Context Invalidation Request Granularity (CIRG):** Software provides the requested invalidation granularity through this field when setting the ICC field: • 00: Reserved. • 01: Global Invalidation request. • 10: Domain-selective invalidation request. The target domain-id must be specified in the DID field. • 11: Device-selective invalidation request. The target source-id(s) must be specified through the SID and FM fields, and the domain-id (that was programmed in the context-entry for these device(s)) must be provided in the DID field. Hardware implementations may process an invalidation request by performing invalidation at a coarser granularity than requested. Hardware indicates completion of the invalidation request by clearing the ICC field. At this time, hardware also indicates the granularity at which the actual invalidation was performed through the CAIG field.

Bit Range	Default & Access	Field Name (ID): Description
60:59	1h RO/V	**Context Actual Invalidation Granularity (CAIG):** Hardware reports the granularity at which an invalidation request was processed through the CAIG field at the time of reporting invalidation completion (by clearing the ICC field). The following are the encodings for this field: • 00: Reserved. • 01: Global Invalidation performed. This could be in response to a global, domain-selective or device-selective invalidation request. • 10: Domain-selective invalidation performed using the domain-id specified by software in the DID field. This could be in response to a domain-selective or device-selective invalidation request. • 11: Device-selective invalidation performed using the source-id and domain-id specified by software in the SID and FM fields. This can only be in response to a device-selective invalidation request. Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
58:34	0h RO	Reserved.
33:32	0h RW	**Function Mask (FM):** Software may use the Function Mask to perform device-selective invalidations on behalf of devices supporting PCI Express Phantom Functions...This field specifies which bits of the function number portion (least significant three bits) of the SID field to mask when performing device-selective invalidations. The following encodings are defined for this field: • 00: No bits in the SID field masked. • 01: Mask most significant bit of function number in the SID field. • 10: Mask two most significant bit of function number in the SID field. • 11: Mask all three bits of function number in the SID field. The context-entries corresponding to all the source-ids specified through the FM and SID fields must have to the domain-id specified in the DID field. Note: This field behaves like a WO field except for requests with certain SAIs: Power Management, DFX Red2, and DFX Red4.
31:16	0h RW	**Source-ID (SID):** Indicates the source-id of the device whose corresponding context-entry needs to be selectively invalidated. This field along with the FM field must be programmed by software for device-selective invalidation requests. Note: This field behaves like a WO field except for requests with certain SAIs: Power Management, DFX Red2, and DFX Red4.
15:0	0h RW	**Domain-ID (DID):** Indicates the id of the domain whose context-entries need to be selectively invalidated. This field must be programmed by software for both domain-selective and device-selective invalidation requests. The Capability register reports the domain-id width supported by hardware. Software must ensure that the value written to this field is within this limit. Hardware may ignore and not implement bits15:N, where N is the supported domain-id width reported in the Capability register.

32.211 Fault Status Register (FSTS_REG_0_0_0_VTDBAR)—Offset 34h

Register indicating the various error status.

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	Reserved.
15:8	0h RO	**Fault Record Index (FRI):** This field is valid only when the PPF field is Set. The FRI field indicates the index (from base) of the fault recording register to which the first pending fault was recorded when the PPF field was Set by hardware. The value read from this field is undefined when the PPF field is clear.
7	0h RW/1C	**Page Request Overflow (PRO):** Hardware detected a Page Request Overflow error. Hardware implementations not supporting the Page Request Queue implement this bit as RsvdZ. Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
6	0h RO	**Invalidation Time-out Error (ITE):** Hardware detected a Device-IOTLB invalidation completion time-out. At this time, a fault event may be generated based on the programming of the Fault Event Control register. Hardware implementations not supporting device Device-IOTLBs implement this bit as RsvdZ.
5	0h RO	**Invalidation Completion Error (ICE):** Hardware received an unexpected or invalid Device-IOTLB invalidation completion. This could be due to either an invalid ITag or invalid source-id in an invalidation completion response. At this time, a fault event may be generated based on the programming of hte Fault Event Control register. Hardware implementations not supporting Device-IOTLBs implement this bit as RsvdZ.
4	0h RW/1C	**Invalidation Queue Error (IQE):** Hardware detected an error associated with the invalidation queue. This could be due to either a hardware error while fetching a descriptor from the invalidation queue, or hardware detecting an erroneous or invalid descriptor in the invalidation queue. At this time, a fault event may be generated based on the programming of the Fault Event Control register. Hardware implementations not supporting queued invalidations implement this bit as RsvdZ. Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
3	0h RO	**Advanced Pending Fault (APF):** When this field is Clear, hardware sets this field when the first fault record (at index 0) is written to a fault log. At this time, a fault event is generated based on the programming of the Fault Event Control register. Software writing 1 to this field clears it. Hardware implementations not supporting advanced fault logging implement this bit as RsvdZ.
2	0h RO	**Advanced Fault Overflow (AFO):** Hardware sets this field to indicate advanced fault log overflow condition. At this time, a fault event is generated based on the programming of the Fault Event Control register. Software writing 1 to this field clears it. Hardware implementations not supporting advanced fault logging implement this bit as RsvdZ.
1	0h RO/V	**Primary Pending Fault (PPF):** This field indicates if there are one or more pending faults logged in the fault recording registers. Hardware computes this field as the logical OR of Fault (F) fields across all the fault recording registers of this remapping hardware unit. • 0 = No pending faults in any of the fault recording registers. • 1 = One or more fault recording registers has pending faults. The FRI field is updated by hardware whenever the PPF field is set by hardware. Also, depending on the programming of Fault Event Control register, a fault event is generated when hardware sets this field. Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
0	0h RW/1C	**Primary Fault Overflow (PFO):** Hardware sets this field to indicate overflow of fault recording registers. Software writing 1 clears this field. When this field is Set, hardware does not record any new faults until software clears this field. Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.

32.212 Fault Event Control Register (FECTL_REG_0_0_0_VTDBAR)—Offset 38h

Register specifying the fault event interrupt message control bits.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 80000000h

Bit Range	Default & Access	Field Name (ID): Description
31	1h RW	**Interrupt Mask (IM):** • 0 = No masking of interrupt. When an interrupt condition is detected, hardware issues an interrupt message (using the Fault Event Data and Fault Event Address register values). • 1 = This is the value on reset. Software may mask interrupt message generation by setting this field. Hardware is prohibited from sending the interrupt message when this field is set.

30	0h
	RO/V
		Interrupt Pending (IP): Hardware sets the IP field whenever it detects an interrupt condition, which is defined as:
		• When primary fault logging is active, an interrupt condition occurs when hardware records a fault through one of the Fault Recording registers and sets the PPF field in Fault Status register.
		• When advanced fault logging is active, an interrupt condition occurs when hardware records a fault in the first fault record (at index 0) of the current fault log and sets the APF field in the Fault Status register.
		• Hardware detected error associated with the Invalidation Queue, setting the IQE field in the Fault Status register.
		• Hardware detected invalid Device-IOTLB invalidation completion, setting the ICE field in the Fault Status register.
		• Hardware detected Device-IOTLB invalidation completion time-out, setting the ITE field in the Fault Status register.
		If any of the status fields in the Fault Status register was already Set at the time of setting any of these fields, it is not treated as a new interrupt condition. The IP field is kept set by hardware while the interrupt message is held pending. The interrupt message could be held pending due to interrupt mask (IM field) being Set or other transient hardware conditions. The IP field is cleared by hardware as soon as the interrupt message pending condition is serviced. This could be due to either:
		• Hardware issuing the interrupt message due to either change in the transient hardware condition that caused interrupt message to be held pending, or due to software clearing the IM field.
		• Software servicing all the pending interrupt status fields in the Fault Status register as follows:
		  • When primary fault logging is active, software clearing the Fault (F) field in all the Fault Recording registers with faults, causing the PPF field in Fault Status register to be evaluated as clear.
		  • Software clearing other status fields in the Fault Status register by writing back the value read from the respective fields.
		Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.

32.213 LJPLL_RW_CONTROL_0 (LJPLL_CR_RW_CONTROL_0)—Offset 0h

LJPLL CR

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 6Eh

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RW	**RESERVED_0**
25	0h RW	**SKIP_FUSE_PULL:** If set Don t repull fuses on side_rst deassertion
24:18	0h RW	**LJPLL_OUTPUT_RATIO:** Set the output clk ratio
17:16	0h RW	**LJPLL_PVD_RATIO:** Set the post VCO clk divider ratio 001 012 104 118
15:14	0h RW	**LJPLL_REF_RATIO:** Set the refclk ratio 00 refclk 01 refclk/2 10 refclk/4 11refclk/8 set this to 2 b10 for Iunit PLL
13	0h RW	**LJPLL_FORCE_ON:** Force The PLL Enable ON 0NoForce 1 FORCE_ON
12	0h RW	**LJPLL_FORCE_OFF:** Force The PLL Enable OFF 0 NoForce 1 FORCE_OFF
11:10	0h RW	**SEL_MIPICLK_C:** Select DSI Clk 00 default 018x 1016x/3 114x don t care for other PLLs
9:8	0h RW	**SEL_MIPICLK_A:** Select DSI Clk 00default 018x 1016x/3 114x don t care for other PLLs
7:0	6Eh RW	**LJPLL_FB_RATIO:** Set the feedback ratio

32.214 LJPLL_RW_CONTROL_1 (LJPLL_CR_RW_CONTROL_1)—Offset 10h

LJPLL CR

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**PLL_RATIO_FRAC:** Clock Bending: fractional frequency multiplier; shift PLL clock frequency by (value/2^24)*refclk frequency. eg 0x200000 = (2097152/2^24) * refclk freq = 0.125*19.2 = 2.4MHz
7:2	0h RW	**SPARE:** Spare CR
1	0h RW	**SSC_EN_OVRD:** Override the fuse/tap value for SSC enable
0	0h RW	**SSC_EN:** Spread Spectrum Clocking: spread enable; 0x0=no frequency spreading; 0x1=enable frequency spreading on PLL output clock; This enable is not default and needs to be set along with SSC_EN_OVRD to take effect

32.215 LJPLL_RW_CONTROL_2 (LJPLL_CR_RW_CONTROL_2)—Offset 14h

LJPLL CR

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**SSC_FRAC_STEP:** Spread Spectrum Clocking: fractional step configuration; fraction of PLL ratio at which to take frequency modulation steps. eg 0x200000 = (2097152/ 2^20) * refclk freq = 0.125*19.2 = 2.4MHz steps. Spread magnitude is determined by the step size multiplied by the number of steps in the modulation period (see ssc_cyc_to_peak_m1 for steps per modulation period).
11	0h RW	**SPARE:** Spare CR
10:9	0h RW	**SSC_MODE:** Spread Spectrum Clocking: spread direction select; 0x0 = down-spread only; 0x1 = up-spread only; 0x2 = center spread, start with down-spread; 0x3 = center spread, start with up-spread
8:0	0h RW	**SSC_CYC_TO_PEAK_M1:** Spread Spectrum Clocking: spread period configuration; half the number of steps in the modulation period minus 1. Period of modulation is 2*(value+1) multiplied by the step duration (PLL refclk period). eg 0x12B = 2*(299+1) * (1/19.2MHz) = 600 * 52.083ns = 31.25us. Spread magnitude is determined by the step size (integer + fractional) multiplied by the number of steps in the modulation period (see ssc_frac_step and ssc_ratio_step for step

32.216 dsipll_cp (dsipll_cp)—Offset 20h

Policy DSIPLL_POLICY_GROUP CP Register

Access Method

Type: MEM Register
(Size: 64 bits)

Device:
Function:

Default: 40001200215h

Bit Range	Default & Access	Field Name (ID): Description
63:0	40001200215h RW	**sai**

32.217 dsipll_rac (dsipll_rac)—Offset 28h

Policy DSIPLL_POLICY_GROUP RAC Register

Access Method

Type: MEM Register
(Size: 64 bits)

Device:
Function:

Default: 40001200215h

Bit Range	Default & Access	Field Name (ID): Description
63:0	40001200215h RW	**sai**

32.218 dsipll_wac (dsipll_wac)—Offset 30h

Policy DSIPLL_POLICY_GROUP WAC Register

Access Method

Type: MEM Register
(Size: 64 bits)

Device:
Function:

Default: 40001200215h

Bit Range	Default & Access	Field Name (ID): Description
63:0	40001200215h RW	**sai**

32.219 LJPLL_RW_CONTROL_0 (LJPLL_CR_RW_CONTROL_0)—Offset 1000h

LJPLL CR

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 4h

Bit Range	Default & Access	Field Name (ID): Description
31:26	0h RW	**RESERVED_0**
25	0h RW	**SKIP_FUSE_PULL:** If set Don t repull fuses on side_rst deassertion
24:18	0h RW	**LJPLL_OUTPUT_RATIO:** Set the output clk ratio
17:16	0h RW	**LJPLL_PVD_RATIO:** Set the post VCO clk divider ratio 001 012 104 118
15:14	0h RW	**LJPLL_REF_RATIO:** Set the refclk ratio 00 refclk 01 refclk/2 10 refclk/4 11refclk/8 set this to 2 b10 for Iunit PLL

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**LJPLL_FORCE_ON:** Force The PLL Enable OFF
12	0h RW	**LJPLL_FORCE_OFF:** Select DSI Clk 00 for non DSI/MIPI instances 018x 1016x/3 114x
11:10	0h RW	**SEL_MIPICLK_C:** Select DSI Clk 00 for non DSI/MIPI instances 018x 1016x/3 114x
9:8	0h RW	**SEL_MIPICLK_A:** Force The PLL Enable ON
7:0	4h RW	**LJPLL_FB_RATIO:** Set the feedback ratio

32.220 LJPLL_RW_CONTROL_1 (LJPLL_CR_RW_CONTROL_1)—Offset 1010h

LJPLL CR

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RW	**PLL_RATIO_FRAC:** Clock Bending: fractional frequency multiplier; shift PLL clock frequency by (value/2^24)*refclk frequency. eg 0x200000 = (2097152/2^24) * refclk freq = 0.125*19.2 = 2.4MHz
7:2	0h RW	**SPARE:** Spare CR
1	0h RW	**SSC_EN_OVRD:** Override the fuse/tap value for SSC enable
0	0h RW	**SSC_EN:** Spread Spectrum Clocking: spread enable; 0x0=no frequency spreading; 0x1=enable frequency spreading on PLL output clock; This enable is not default and needs to be set along with SSC_EN_OVRD to take effect

32.221 LJPLL_RW_CONTROL_2 (LJPLL_CR_RW_CONTROL_2)—Offset 1014h

LJPLL CR

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:12	0h RW	**SSC_FRAC_STEP:** Spread Spectrum Clocking: fractional step configuration; fraction of PLL ratio at which to take frequency modulation steps. eg 0x200000 = (2097152/ 2^20) * refclk freq = 0.125*19.2 = 2.4MHz steps. Spread magnitude is determined by the step size multiplied by the number of steps in the modulation period (see ssc_cyc_to_peak_m1 for steps per modulation period).
11	0h RW	**SPARE:** Spare CR
10:9	0h RW	**SSC_MODE:** Spread Spectrum Clocking: spread direction select; 0x0 = down-spread only; 0x1 = up-spread only; 0x2 = center spread, start with down-spread; 0x3 = center spread, start with up-spread
8:0	0h RW	**SSC_CYC_TO_PEAK_M1:** Spread Spectrum Clocking: spread period configuration; half the number of steps in the modulation period minus 1. Period of modulation is 2*(value+1) multiplied by the step duration (PLL refclk period). eg 0x12B = 2*(299+1) * (1/19.2MHz) = 600 * 52.083ns = 31.25us. Spread magnitude is determined by the step size (integer + fractional) multiplied by the number of steps in the modulation period (see ssc_frac_step and ssc_ratio_step for step

Version Register (VER_REG_0_0_0_VTDBAR)—Offset 0h

Register to report the architecture version supported. Backward compatibility for the architecture is maintained with new revision numbers, allowing software to load remapping hardware drivers written for prior architecture versions.

Access Method

Type: MEM Register (Size: 32 bits)
Device:
Function:

Default: 10h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	Reserved.
7:4	1h RO	**Major Version Number (MAJOR):** Indicates supported architecture version.
3:0	0h RO	**Minor Version Number (MINOR):** Indicates supported architecture minor version.

32.222 Capability Register (CAP_REG_0_0_0_VTDBAR)—Offset 8h

Register to report general remapping hardware capabilities. Note: These values can change based on defeature bits.

Access Method

Type: MEM Register (Size: 64 bits)
Device:
Function:

Default: D2008C40660462h

Bit Range	Default & Access	Field Name (ID): Description
63:57	0h RO	Reserved.
56	0h RO	**First Level 64-KByte Page SupportP (FL1GP):** A value of 1 in this field indicates 1-GByte page size is supported for first-level translation. Note: These values can change based on defeature bits.
55	1h RO	**Read Draining (DRD):** • 0 = Hardware does not support draining of DMA read requests. • 1 = Hardware supports draining of DMA read requests. Note: These values can change based on defeature bits.
54	1h RO	**Write Draining (DWD):** • 0 = Hardware does not support draining of DMA write requests. • 1 = Hardware supports draining of DMA write requests. Note: These values can change based on defeature bits.
53:48	12h RO	**Maximum Address Mask Value (MAMV):** The value in this field indicates the maximum supported value for the Address Mask (AM) field in the Invalidation Address register (IVA_REG) and IOTLB Invalidation Descriptor (iotlb_inv_dsc) used for invalidations of second-level translation. This field is valid only when the PSI field in Capability register is reported as Set. Note: These values can change based on defeature bits.
47:40	0h RO	**Number of Fault-Recording Registers (NFR):** Number of fault recording registers is computed as N+1, where N is the value reported in this field. Implementations must support at least one fault recording register (NFR = 0) for each remapping hardware unit in the platform. The maximum number of fault recording registers per remapping hardware unit is 256. Note: These values can change based on defeature bits.
39	1h RO	**Page Selective Invalidation (PSI):** • 0 = Hardware supports only domain and global invalidates for IOTLB.{*}1 = Hardware supports page selective, domain and global invalidates for IOTLB. Hardware implementations reporting this field as set are recommended to support a Maximum Address Mask Value (MAMV) value of at least 9 (or 18 if supporting 1GB pages with second level translation). Note: These values can change based on defeature bits.
38	0h RO	Reserved.
37:34	3h RO	**Second Level Large Page Support (SLLPS):** This field indicates the super page sizes supported by hardware. A value of 1 in any of these bits indicates the corresponding super-page size is supported. The super-page sizes corresponding to various bit positions within this field are: • 0 = 21-bit offset to page frame (2MB) • 1 = 30-bit offset to page frame (1GB) • 2 = 39-bit offset to page frame (512GB) • 3 = 48-bit offset to page frame (1TB) Hardware implementations supporting a specific super-page size must support all smaller super-page sizes, i.e. only valid values for this field are 0000b, 0001b, 0011b, 0111b, 1111b. Note: These values can change based on defeature bits.
33:24	40h RO	**Fault-Recording Register Offset (FRO):** This field specifies the location to the first fault recording register relative to the register base address of this remapping hardware unit. If the register base address is X, and the value reported in this field is Y, the address for the first fault recording register is calculated as X+(16*Y). Note: These values can change based on defeature bits.
23	0h RO	Reserved.

Bit Range	Default & Access	Field Name (ID): Description
22	1h RO	**Zero Length Read (ZLR):** • 0 = Indicates the remapping hardware unit blocks (and treats as fault) zero length DMA read requests to write-only pages. • 1 = Indicates the remapping hardware unit supports zero length DMA read requests to write-only pages. DMA remapping hardware implementations are recommended to report ZLR field as Set. Note: These values can change based on defeature bits.
21:16	26h RO	**Maximum Guest Address Width (MGAW):** This field indicates the maximum DMA virtual addressability supported by remapping hardware. The Maximum Guest Address Width (MGAW) is computed as (N+1), where N is the value reported in this field. For example, a hardware implementation supporting 48-bit MGAW reports a value of 47 (101111b) in this field. If the value in this field is X, untranslated and translated DMA requests to addresses above 2(x+1)-1 are always blocked by hardware. Translations requests to address above 2(x+1)-1 from allowed devices return a null Translation Completion Data Entry with R=W=0. Guest addressability for a given DMA request is limited to the minimum of the value reported through this field and the adjusted guest address width of the corresponding page-table structure. (Adjusted guest address widths supported by hardware are reported through the SAGAW field). Implementations are recommended to support MGAW at least equal to the physical addressability (host address width) of the platform. Note: These values can change based on defeature bits.
15:13	0h RO	Reserved.
12:8	4h RO	**Supported Adjusted Guest Address Widths (SAGAW):** This 5-bit field indicates the supported adjusted guest address widths (which in turn represents the levels of page-table walks for the 4KB base page size) supported by the hardware implementation. A value of 1 in any of these bits indicates the corresponding adjusted guest address width is supported. The adjusted guest address widths corresponding to various bit positions within this field are: • 0 = 30-bit AGAW (2-level page table) • 1 = 39-bit AGAW (3-level page table) • 2 = 48-bit AGAW (4-level page table) • 3 = 57-bit AGAW (5-level page table) • 4 = 64-bit AGAW (6-level page table) Software must ensure that the adjusted guest address width used to setup the page tables is one of the supported guest address widths reported in this field. Note: These values can change based on defeature bits.
7	0h RO	**Caching Mode (CM):** • 0 = Not-present and erroneous entries are not cached in any of the renmapping caches. Invalidations are not required for modifications to individual not present or invalid entries. However, any modifications that result in decreasing the effective permissions or partial permission increases require invalidations for them to be effective. • 1 = Not-present and erroneous mappings may be cached in the remapping caches. Any software updates to the remapping structures (including updates to not-present or erroneous entries) require explicit invalidation. Hardware implementations of this architecture must support a value of 0 in this field. Note: These values can change based on defeature bits.
6	1h RO	**Protected High-Memory Region (PHMR):** • 0 = Indicates protected high-memory region is not supported. • 1 = Indicates protected high-memory region is supported. Note: These values can change based on defeature bits.
5	1h RO	**Protected Low-Memory Region (PLMR):** • 0 = Indicates protected low-memory region is not supported. • 1 = Indicates protected low-memory region is supported. Note: These values can change based on defeature bits.

Bit Range	Default & Access	Field Name (ID): Description
4	0h RO	**Required Write-Buffer Flushing (RWBF):** • 0 = Indicates no write-buffer flushing is needed to ensure changes to memory-resident structures are visible to hardware. • 1 = Indicates software must explicitly flush the write buffers to ensure updates made to memory-resident remapping structures are visible to hardware. Note: These values can change based on defeature bits.
3	0h RO	**Advanced Fault Logging (AFL):** • 0 = Indicates advanced fault logging is not supported. Only primary fault logging is supported. • 1 = Indicates advanced fault logging is supported. Note: These values can change based on defeature bits.
2:0	2h RO	**Number of Domains Supported (ND):** • 000b = Hardware supports 4-bit domain-ids with support for up to 16 domains. • 001b = Hardware supports 6-bit domain-ids with support for up to 64 domains. • 010b = Hardware supports 8-bit domain-ids with support for up to 256 domains. • 011b = Hardware supports 10-bit domain-ids with support for up to 1024 domains. • 100b = Hardware supports 12-bit domain-ids with support for up to 4K domains. • 100b = Hardware supports 14-bit domain-ids with support for up to 16K domains. • 110b = Hardware supports 16-bit domain-ids with support for up to 64K domains. • 111b = Reserved. Note: These values can change based on defeature bits.

32.223 Extended Capability Register (ECAP_REG_0_0_0_VTDBAR)—Offset 10h

Register to report remapping hardware extended capabilities. Note: These values can change based on defeature bits.

Access Method

Type: MEM Register
(Size: 64 bits)

Device:
Function:

Default: F050DAh

Bit Range	Default & Access	Field Name (ID): Description
63:41	0h RO	Reserved.
40	0h RO	**Process Address Space ID Support (PASID):** • 0 = Hardware does not support requests tagged with Process Address Space IDs. • 1 = Hardware supports requests tagged with Process Address Space IDs. Note: These values can change based on defeature bits.
39:35	0h RO	**PASID Size Supported (PSS):** This field reports the PASID size supported by the remapping hardware for requests-with-PASID. A value of N in this field indicates hardware supports PASID field of N+1 bits (For example, value of 7 in this field, indicates 8-bit PASIDs are supported). Requests-with-PASID with PASID value beyond the limit specified by this field are treated as error by the remapping hardware. This field is valid only when PASID field is reported as Set. Note: These values can change based on defeature bits.

Bit Range	Default & Access	Field Name (ID): Description
34	0h RO	**Extended Accessed Flag Support (EAFS):** • 0 = Hardware does not support the extended-accessed (EA) bit in first-level paging-structure entries. • 1 = Hardware supports the extended accessed (EA) bit in first-level paging-structure entries. This field is valid only when PASID field is reported as Set. Note: These values can change based on defeature bits.
33	0h RO	**No Write Flag Support (NWFS):** • 0 = Hardware ignores the No Write (NW) flag in Device-TLB translationrequests, and behaves as if NW is always 0. • 1 = Hardware supports the No Write (NW) flag in Device-TLB translationrequests. This field is valid only when Device-TLB support (DT) field is reported as Set. Note: These values can change based on defeature bits.
32	0h RO	**PASID-Only Translations (POT):** • 0 = Hardware does not support PASID-only Translation Type in extended-context-entries. • 1 = Hardware supports PASID-only Translation Type in extended-context-entries. This field is valid only when PASID field is reported as Set. Note: These values can change based on defeature bits.
31	0h RO	**Supervisor Request Support (SRS):** • 0 = H/W does not support requests-with-PASID seeking supervisor privilege. • 1 = H/W supports requests-with-PASID seeking supervisor privilege. The field is valid only when PASID field is reported as Set. Note: These values can change based on defeature bits.
30	0h RO	**Execute Request Support (ERS):** • 0 = H/W does not support requests-with-PASID seeking execute permission. • 1 = H/W supports requests-with-PASID seeking execute permission. This field is valid only when PASID field is reported as Set. Note: These values can change based on defeature bits.
29	0h RO	**Page Request Support (PRS):** • 0 = Hardware does not support Page Requests. • 1 = Hardware supports Page Requests This field is valid only when Device-TLB (DT) field is reported as Set. Note: These values can change based on defeature bits.
28	0h RO	**Ignore (IGN):** Ignore this field
27	0h RO	**Deferred Invalidate Support (DIS):** • 0 = Hardware does not support deferred invalidations of IOTLB and Device-TLB. • 1 = Hardware supports deferred invalidations of IOTLB and Device-TLB. This field is valid only when PASID field is reported as Set. Note: These values can change based on defeature bits.
26	0h RO	**Nested Translation Support (NEST):** • 0 = Hardware does not support nested translations. • 1 = Hardware supports nested translations. This field is valid only when PASID field is reported as Set. Note: These values can change based on defeature bits.
25	0h RO	**Memory Type Support (MTS):** • 0 = Hardware does not support Memory Type in first-level translation and Extended Memory type in second-level translation. • 1 = Hardware supports Memory Type in first-level translation and Extended Memory type in second-level translation. This field is valid only when PASID and ECS fields are reported as Set. Remapping hardware units with, one or more devices that operate in processor coherency domain, under its scope must report this field as Set. Note: These values can change based on defeature bits.

Bit Range	Default & Access	Field Name (ID): Description
24	0h RO	**Extended Context Support (ECS):** • 0 = Hardware does not support extended-root-entries and extended-context-entries. • 1 = Hardware supports extended-root-entries and extended-context-entries. Implementations reporting PASID or PRS fields as Set, must report this field as Set. Note: These values can change based on defeature bits.
23:20	Fh RO	**Maximum Handle Mask Value (MHMV):** The value in this field indicates the maximum supported value for the Handle Mask (HM) field in the interrupt entry cache invalidation descriptor (iec_inv_dsc). This field is valid only when the IR field in Extended Capability register is reported as Set. Note: These values can change based on defeature bits.
19:18	0h RO	Reserved.
17:8	50h RO	**IOTLB Register Offset (IRO):** This field specifies the offset to the IOTLB registers relative to the register base address of this remapping hardware unit. If the register base address is X, and the value reported in this field is Y, the address for the first IOTLB invalidation register is calculated as X+(16*Y). Note: These values can change based on defeature bits.
7	1h RO	**Snoop Control (SC):** • 0 = Hardware does not support 1-setting of the SNP field in the page-table entries. • 1 = Hardware supports the 1-setting of the SNP field in the page-table entries. Note: These values can change based on defeature bits.
6	1h RO	**Pass Through (PT):** • 0 = Hardware does not support pass-through translation type in context entries and extended-context-entries. • 1 = Hardware supports pass-through translation type in context entries and extended-context-entries. Pass-through translation is specified through Translation-Type (T) field value of 10b in context-entries, or T field value of 010b in extended-context-entries. Hardware implementations supporting PASID must report a value of 1b in this field. Note: These values can change based on defeature bits.
5	0h RO	Reserved.
4	1h RO	**Extended Interrupt Mode (EIM):** • 0 = On Intel64 platforms, hardware supports only 8-bit APIC-IDs (xAPIC mode). • 1 = On Intel64 platforms, hardware supports 32-bit APIC-IDs (x2APIC mode). This field is valid only on Intel64 platforms reporting Interrupt Remapping support (IR field Set). Note: These values can change based on defeature bits.
3	1h RO	**Interrupt Remapping Suport (IR):** • 0 = Hardware does not support interrupt remapping. • 1 = Hardware supports interrupt remapping. Implementations reporting this field as Set must also support Queued Invalidation (QI). Note: These values can change based on defeature bits.
2	0h RO	**Device-TLB Support (DT):** • 0 = Hardware does not support device-IOTLBs. • 1 = Hardware supports Device-IOTLBs. Implementations reporting this field as Set must also support Queued Invalidation (QI). Hardware implementations supporting I/O Page Requests (PRS field Set in Extended Capability register) must report a value of 1b in this field. Note: These values can change based on defeature bits.
1	1h RO	**Queued Invalidation Support (QI):** • 0 = Hardware does not support queued invalidations. • 1 = Hardware supports queued invalidations. Note: These values can change based on defeature bits.

Bit Range	Default & Access	Field Name (ID): Description
0	0h RO	**Page-Walk Coherency (C):** This field indicates if hardware access to the root, context, extended-context and interrupt-remap tables, and second-level paging structures for requests-without-PASID, are coherent (snooped) or not. • 0 = Indicates hardware accesses to remapping structures are non-coherent. • 1 = Indicates hardware accesses to remapping structures are coherent. Hardware access to advanced fault log, invalidation queue, invalidation semaphore, page-request queue, PASID-table, PASID-state table, and first-level page-tables are always coherent. Note: These values can change based on defeature bits.

32.224 Global Command Register (GCMD_REG_0_0_0_VTDBAR)—Offset 18h

Register to control remapping hardware. If multiple control fields in this register need to be modified, software must serialize the modifications through multiple writes to this register.

Access Method

Type: MEM Register (Size: 32 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Translation Enable (TE):** Software writes to this field to request hardware to enable/disable DMA-remapping: • 0 = Disable DMA remapping. • 1 = Enable DMA remapping. Hardware reports the status of the translation enable operation through the TES field in the Global Status register. There may be active DMA requests in the platform when software updates this field. Hardware must enable or disable remapping logic only at determinstic transaction boundaries, so that any in-flight transaction is either subject to remapping or not at all. Hardware implementations supporting DMA draining must drain any in-flight DMA read/write requests queued within the Root-Complex before completing the translation enable command and reflecting the status of the command through the TES field in the Global Status register. The value returned on a read of this field is undefined. Note: This field behaves like a WO field except for requests with certain SAIs: Power Management, DFX Red2, and DFX Red4.
30	0h WO	**Set Root Table Pointer (SRTP):** Software sets this field to set/update the root-entry table pointer used by hardware. The root-entry table pointer is specified through the Root-entry Table Address (RTA_REG) register. Hardware reports the status of the Set Root Table Pointer operation through the RTPS field in the Global Status register. The Set Root Table Pointer operation must be performed before enabling or re-enabling (after disabling) DMA remapping through the TE field. .After a Set Root Table Pointer operation, software must globally invalidate the context cache and then globally invalidate of IOTLB. This is required to ensure hardware uses only the remapping structures referenced by the new root table pointer, and not stale cached entries. While DMA remapping hardware is active, software may update the root table pointer through this field. However, to ensure valid in-flight DMA requests are deterministically remapped, software must ensure that the structures referenced by the new root table pointer are programmed to provide the same remapping results as the structures referenced by the previous root-table pointer. Clearing this bit has no effect. The value returned on read of this field is undefined. Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.

Bit Range	Default & Access	Field Name (ID): Description
29	0h RO	**Set Fault Log (SFL):** This field is valid only for implementations supporting advanced fault logging. Software sets this field to request hardware to set/update the fault-log pointer used by hardware. The fault-log pointer is specified through Advanced Fault Log register. Hardware reports the status of the Set Fault Log operation through the FLS field in the Global Status register. The fault log pointer must be set before enabling advanced fault logging (through EAFL field). Once advanced fault logging is enabled, the fault log pointer may be updated through this field while DMA remapping is active. Clearing this bit has no effect. The value returned on read of this field is undefined.
28	0h RO	**Enable Advanced Fault Logging (EAFL):** This field is valid only for implementations supporting advanced fault logging. Software writes to this field to request hardware to enable or disable advanced fault logging: • 0 = Disable advanced fault logging. In this case, translation faults are reported through the Fault Recording registers. • 1 = Enable use of memory-resident fault log. When enabled, translation faults are recorded in the memory-resident log. The fault log pointer must be set in hardware (through the SFL field) before enabling advaned fault logging. Hardware reports the status of the advaned fault logging enable operation through the AFLS field in the Global Status register. The value returned on read of this field is undefined.
27	0h RO	**Write Buffer Flush (WBF):** This bit is valid only for implementations requiring write buffer flushing. Software sets this field to request that hardware flush the Root-Complex internal write buffers. This is done to ensure any updates to the memory-resident remapping structures are not held in any internal write posting buffers. Hardware reports the status of the write buffer flushing operation through the WBFS field in the Global Status register. Clearing this bit has no effect. The value returned on a read of this field is undefined.
26	0h RW	**Queued Invalidation Enable (QIE):** This field is valid only for implementations supporting queued invalidations. Software writes to this field to enable or disable queued invalidations. • 0 = Disable queued invalidations. • 1 = Enable use of queued invalidations. Hardware reports the status of queued invalidation enable operation through QIES field in the Global Status register. The value returned on a read of this field is undefined. Note: This field behaves like a WO field except for requests with certain SAIs: Power Management, DFX Red2, and DFX Red4.
25	0h RW	**Interrupt Remapping Enable (IRE):** This field is valid only for implementations supporting interrupt remapping. • 0 = Disable interrupt-remapping hardware. • 1 = Enable interrupt-remapping hardware. Hardware reports the status of the interrupt remapping enable operation through the IRES field in the Global Status register. There may be active interrupt requests in the platform when software updates this field. Hardware must enable or disable interrupt-remapping logic only at deterministic transaction boundaries, so that any in-flight interrupts are either subject to remapping or not at all. Hardware implementations must drain any in-flight interrupts requests queued in the Root-Complex before completing the interrupt-remapping enable command and reflecting the status of the command through the IRES field in the Global Status register. The value returned on a read of this field is undefined. Note: This field behaves like a WO field except for requests with certain SAIs: Power Management, DFX Red2, and DFX Red4.

Bit Range	Default & Access	Field Name (ID): Description
24	0h WO	**Set Interrupt Remap Table Pointer (SIRTP):** This field is valid only for implementations supporting interrupt-remapping. Software sets this field to set/update the interrupt remapping table pointer used by hardware. The interrupt remapping table pointer is specified through the Interrupt Remapping Table Address (IRTA_REG) register. Hardware reports the status of the Set Interrupt Remap Table Pointer operation through the IRTPS field in the Global Status register. The Set Interrupt Remap Table Pointer operation must be performed before enabling or re-enabling (after disabling) interrupt-remapping hardware through the IRE field. After a Set Interrupt Remap Table Pointer operation, software must globally invalidate the interrupt entry cache. This is required to ensure hardware uses only the interrupt-remapping entries referenced by the new interrupt remap table pointer, and not any stale cached entries. While interrupt remapping is active, software may update the interrupt remapping table pointer through this field. However, to ensure valid in-flight interrupt requests are deterministically remapped, software must ensure that the structures referenced by the new interrupt remap table pointer are programmed to provide the same remapping results as the structures referenced by the previous interrupt remap table pointer. Clearing this bit has no effect. The value returned on a read of this field is undefined. Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
23	0h RW	**Compatibility Format Interrupt (CFI):** This field is valid only for Intel64 implementations supporting interrupt-remapping. Software writes to this field to enable or disable Compatibility Format interrupts on Intel64 platforms. The value in this field is effective only when interrupt-remapping is enabled and Extended Interrupt Mode (x2APIC mode) is not enabled. • 0 = Block Compatibility format interrupts. • 1 = Process Compatibility format interrupts as pass-through (bypass interrupt remapping). Hardware reports the status of updating this field through the CFIS field in the Global Status register. The value returned on a read of this field is undefined. Note: This field behaves like a WO field except for requests with certain SAIs: Power Management, DFX Red2, and DFX Red4.
22:0	0h RO	Reserved.

32.225 Global Status Register (GSTS_REG_0_0_0_VTDBAR)—Offset 1Ch

Register to report general remapping hardware status.

Access Method

Type: MEM Register (Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RO/V	**Translation Enable Status (TES):** This field indicates the status of DMA-remapping hardware. • 0 = DMA-remapping hardware is not enabled. • 1 = DMA-remapping hardware is enabled Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.

Bit Range	Default & Access	Field Name (ID): Description
30	0h RO/V	**Root Table Pointer Status (RTPS):** This field indicates the status of the root- table pointer in hardware. This field is cleared by hardware when software sets the SRTP field in the Global Command register. This field is set by hardware when hardware completes the Set Root Table Pointer operation using the value provided in the Root-Entry Table Address register. Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
29	0h RO	**Fault Log Status (FLS):** This field: • Is cleared by hardware when software Sets the SFL field in the Global Command register. • Is Set by hardware whn hardware completes the Set Fault Log Pointer operation using the value provided in the Advanced Fault Log register.
28	0h RO	**Adavanced Fault Logging Status (AFLS):** This field is valid only for implementations supporting advanced fault logging. It indicates the advanced fault logging status: • 0 = Advanced Fault Logging is not enabled. • 1 = Advanced Fault Logging is enabled.
27	0h RO	**Write Buffer Flush Status (WBFS):** This field is valid only for implementations requiring write buffer flushing. This field indicates the status of the write buffer flush command. It is: • Set by hardware when software sets the WBF field in the Global Command register. • Cleared by hardware when hardware completes the write buffer flushing operation.
26	0h RO/V	**Queued Invalidation Enable Status (QIES):** This field indicates queued invalidation enable status. • 0 = queued invalidation is not enabled. • 1 = queued invalidation is enabled Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
25	0h RO/V	**Interrupt Remapping Enable Status (IRES):** This field indicates the status of Interrupt-remapping hardware. • 0 = Interrupt-remapping hardware is not enabled. • 1 = Interrupt-remapping hardware is enabled Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
24	0h RO/V	**Interrupt Remapping Pointer Status (IRTPS):** This field indicates the status of the interrupt remapping table pointer in hardware. This field is cleared by hardware when software sets the SIRTP field in the Global Command register. This field is Set by hardware when hardware completes the set interrupt remap table pointer operation using the value provided in the Interrupt Remapping Table Address register. Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
23	0h RO/V	**Compatibility Format Interrupt Status (CFIS):** This field indicates the status of Compatibility format interrupts on Intel64 implementations supporting interrupt-remapping. The value reported in this field is applicable only when interrupt-remapping is enabled and Extended Interrupt Mode (x2APIC mode) is not enabled. • 0 = Compatibility format interrupts are blocked. • 1 = Compatibility format interrupts are processed as pass-through (bypassing interrupt remapping). Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
22:0	0h RO	Reserved.

32.226 Root Table Address Register (RTADDR_REG_0_0_0_VTDBAR)—Offset 20h

Register providing the base address of root-entry table.

Access Method

Type: MEM Register (Size: 64 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
63:12	0h RW	**Root Table Address (RTA):** This register points to base of page aligned, 4KB-sized root-entry table in system memory. Hardware ignores and not implements bits 63:HAW, where HAW is the host address width. Software specifies the base address of the root-entry table through this register, and programs it in hardware through the SRTP field in the Global Command register. Reads of this register returns value that was last programmed to it.
11	0h RW	**Root Table Type (RTT):** This field specifies the type of root-table referenced by the Root Table Address (RTA) field: • 0 = Root Table. • 1 = Extended Root Table
10:0	0h RO	Reserved.

32.227 Context Command Register (CCMD_REG_0_0_0_VTDBAR)—Offset 28h

Register to manage context cache. The act of writing the uppermost byte of the CCMD_REG with the ICC field Set causes the hardware to perform the context-cache invalidation.

Access Method

Type: MEM Register (Size: 64 bits)
Device:
Function:

Default: 8000000000000000h

Bit Range	Default & Access	Field Name (ID): Description
63	0h RW/V	**Invalidate Context Cache (ICC):** Software requests invalidation of context-cache by setting this field. Software must also set the requested invalidation granularity by programming the CIRG field. Software must read back and check the ICC field is Clear to confirm the invalidation is complete. Software must not update this register when this field is set. Hardware clears the ICC field to indicate the invalidation request is complete. Hardware also indicates the granularity at which the invalidation operation was performed through the CAIG field. Software must submit a context-cache invalidation request through this field only when there are no invalidation requests pending at this remapping hardware unit. Since information from the context-cache may be used by hardware to tag IOTLB entries, software must perform domain-selective (or global) invalidation of IOTLB after the context cache invalidation has completed. Hardware implementations reporting write-buffer flushing requirement (RWBF=1 in Capability register) must implicitly perform a write buffer flush before invalidating the context cache.
62:61	0h RW	**Context Invalidation Request Granularity (CIRG):** Software provides the requested invalidation granularity through this field when setting the ICC field: • 00: Reserved. • 01: Global Invalidation request. • 10: Domain-selective invalidation request. The target domain-id must be specified in the DID field. • 11: Device-selective invalidation request. The target source-id(s) must be specified through the SID and FM fields, and the domain-id (that was programmed in the context-entry for these device(s)) must be provided in the DID field. Hardware implementations may process an invalidation request by performing invalidation at a coarser granularity than requested. Hardware indicates completion of the invalidation request by clearing the ICC field. At this time, hardware also indicates the granularity at which the actual invalidation was performed through the CAIG field.
60:59	1h RO/V	**Context Actual Invalidation Granularity (CAIG):** Hardware reports the granularity at which an invalidation request was processed through the CAIG field at the time of reporting invalidation completion (by clearing the ICC field). The following are the encodings for this field: • 00: Reserved. • 01: Global Invalidation performed. This could be in response to a global, domain-selective or device-selective invalidation request. • 10: Domain-selective invalidation performed using the domain-id specified by software in the DID field. This could be in response to a domain-selective or device-selective invalidation request. • 11: Device-selective invalidation performed using the source-id and domain-id specified by software in the SID and FM fields. This can only be in response to a device-selective invalidation request. Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
58:34	0h RO	Reserved.
33:32	0h RW	**Function Mask (FM):** Software may use the Function Mask to perform device-selective invalidations on behalf of devices supporting PCI Express Phantom Functions...This field specifies which bits of the function number portion (least significant three bits) of the SID field to mask when performing device-selective invalidations. The following encodings are defined for this field: • 00: No bits in the SID field masked. • 01: Mask most significant bit of function number in the SID field. • 10: Mask two most significant bit of function number in the SID field. • 11: Mask all three bits of function number in the SID field. The context-entries corresponding to all the source-ids specified through the FM and SID fields must have to the domain-id specified in the DID field. Note: This field behaves like a WO field except for requests with certain SAIs: Power Management, DFX Red2, and DFX Red4.

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RW	**Source-ID (SID):** Indicates the source-id of the device whose corresponding context-entry needs to be selectively invalidated. This field along with the FM field must be programmed by software for device-selective invalidation requests. Note: This field behaves like a WO field except for requests with certain SAIs: Power Management, DFX Red2, and DFX Red4.
15:0	0h RW	**Domain-ID (DID):** Indicates the id of the domain whose context-entries need to be selectively invalidated. This field must be programmed by software for both domain-selective and device-selective invalidation requests. The Capability register reports the domain-id width supported by hardware. Software must ensure that the value written to this field is within this limit. Hardware may ignore and not implement bits15:N, where N is the supported domain-id width reported in the Capability register.

32.228 Fault Status Register (FSTS_REG_0_0_0_VTDBAR)—Offset 34h

Register indicating the various error status.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:16	0h RO	Reserved.
15:8	0h RO	**Fault Record Index (FRI):** This field is valid only when the PPF field is Set. The FRI field indicates the index (from base) of the fault recording register to which the first pending fault was recorded when the PPF field was Set by hardware. The value read from this field is undefined when the PPF field is clear.
7	0h RW/1C	**Page Request Overflow (PRO):** Hardware detected a Page Request Overflow error. Hardware implementations not supporting the Page Request Queue implement this bit as RsvdZ. Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
6	0h RO	**Invalidation Time-out Error (ITE):** Hardware detected a Device-IOTLB invalidation completion time-out. At this time, a fault event may be generated based on the programming of the Fault Event Control register. Hardware implementations not supporting device Device-IOTLBs implement this bit as RsvdZ.
5	0h RO	**Invalidation Completion Error (ICE):** Hardware received an unexpected or invalid Device-IOTLB invalidation completion. This could be due to either an invalid ITag or invalid source-id in an invalidation completion response. At this time, a fault event may be generated based on the programming of hte Fault Event Control register. Hardware implementations not supporting Device-IOTLBs implement this bit as RsvdZ.
4	0h RW/1C	**Invalidation Queue Error (IQE):** Hardware detected an error associated with the invalidation queue. This could be due to either a hardware error while fetching a descriptor from the invalidation queue, or hardware detecting an erroneous or invalid descriptor in the invalidation queue. At this time, a fault event may be generated based on the programming of the Fault Event Control register. Hardware implementations not supporting queued invalidations implement this bit as RsvdZ. Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RO	**Advanced Pending Fault (APF):** When this field is Clear, hardware sets this field when the first fault record (at index 0) is written to a fault log. At this time, a fault event is generated based on the programming of the Fault Event Control register. Software writing 1 to this field clears it. Hardware implementations not supporting advanced fault logging implement this bit as RsvdZ.
2	0h RO	**Advanced Fault Overflow (AFO):** Hardware sets this field to indicate advanced fault log overflow condition. At this time, a fault event is generated based on the programming of the Fault Event Control register. Software writing 1 to this field clears it. Hardware implementations not supporting advanced fault logging implement this bit as RsvdZ.
1	0h RO/V	**Primary Pending Fault (PPF):** This field indicates if there are one or more pending faults logged in the fault recording registers. Hardware computes this field as the logical OR of Fault (F) fields across all the fault recording registers of this remapping hardware unit. • 0 = No pending faults in any of the fault recording registers. • 1 = One or more fault recording registers has pending faults. The FRI field is updated by hardware whenever the PPF field is set by hardware. Also, depending on the programming of Fault Event Control register, a fault event is generated when hardware sets this field. Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.
0	0h RW/1C	**Primary Fault Overflow (PFO):** Hardware sets this field to indicate overflow of fault recording registers. Software writing 1 clears this field. When this field is Set, hardware does not record any new faults until software clears this field. Note: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.

32.229 Fault Event Control Register (FECTL_REG_0_0_0_VTDBAR)—Offset 38h

Register specifying the fault event interrupt message control bits.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 80000000h

Bit Range	Default & Access	Field Name (ID): Description
31	1h RW	**Interrupt Mask (IM):** • 0 = No masking of interrupt. When an interrupt condition is detected, hardware issues an interrupt message (using the Fault Event Data and Fault Event Address register values). • 1 = This is the value on reset. Software may mask interrupt message generation by setting this field. Hardware is prohibited from sending the interrupt message when this field is set.

Bit Range	Default & Access	Field Name (ID): Description
30	0h RO/V	**Interrupt Pending (IP):** Hardware sets the IP field whenever it detects an interrupt condition, which is defined as: • When primary fault logging is active, an interrupt condition occurs when hardware records a fault through one of the Fault Recording registers and sets the PPF field in Fault Status register. • When advanced fault logging is active, an interrupt condition occurs when hardware records a fault in the first fault record (at index 0) of the current fault log and sets the APF field in the Fault Status register. • Hardware detected error associated with the Invalidation Queue, setting the IQE field in the Fault Status register. • Hardware detected invalid Device-IOTLB invalidation completion, setting the ICE field in the Fault Status register. • Hardware detected Device-IOTLB invalidation completion time-out, setting the ITE field in the Fault Status register. If any of the status fields in the Fault Status register was already Set at the time of setting any of these fields, it is not treated as a new interrupt condition. The IP field is kept set by hardware while the interrupt message is held pending. The interrupt message could be held pending due to interrupt mask (IM field) being Set or other transient hardware conditions. The IP field is cleared by hardware as soon as the interrupt message pending condition is serviced. This could be due to either: • Hardware issuing the interrupt message due to either change in the transient hardware condition that caused interrupt message to be held pending, or due to software clearing the IM field. • Software servicing all the pending interrupt status fields in the Fault Status register as follows: — When primary fault logging is active, software clearing the Fault (F) field in all the Fault Recording registers with faults, causing the PPF field in Fault Status register to be evaluated as clear. — Software clearing other status fields in the Fault Status register by writing back the value read from the respective fields. **Note**: This field becomes a RW field based on request SAIs: Power Managment, DFX Red2, and DFX Red4.

§ §

33 USB 2 Tuning Registers

33.1 c73usb280_USB2 PER PORT (USB2_PER_PORT_PP0)—Offset 4100h

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 5DA81h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Per-Port UTMI_CLK Output Clock Gating Enable (PPUOCGE):** 1: Dynamic Clock Gating is enabled based on L1/L2 suspend indicator. This is the recommended setting for SYNOPSYS based controllers. 0: Output UTMI_CLK is running unless USB PLL is shutdown. This is the recommended setting for FRESCO based controllers.
30	0h RW	**RX EN AFE Control (RXENAC):** 1: Enables the dynamic RX EN behavior. This allows compensation code updates at various boundaries. 0: Disables the dynamic RX EN behavior (function of TXVALID only). RXENAC, RXBIASAC: 00: Legacy mode 01: Illegal 10: Allows additional code update window 11: Allows additional code update window with power saving
29	0h RW	**RX BIAS AFE Control (RXBIASAC):** 1: Enables the AFE bias control 0: Disables the AFE bias control This bit is only valid when internal RXBIAS policy is selected (SELSBRXBIASEN = 0b).
28	0h RW	**TX BIAS AFE Control (TXBIASAC):** 1: Enables the AFE bias control 0: Disables the AFE bias control This bit is only valid when internal TXBIAS policy is selected (SELSBTXBIASEN = 0b).
27	0h RW	**RX CLKEN AFE Clock Gating (RXCLKENACG):** 1: Enables the AFE clock gating 0: Disables the AFE clock gating This bit is only valid when internal RXCLKEN policy is selected (SELRXCLKEN = 0b).
26	0h RW	**TX CLKEN AFE Clock Gating (TXCLKENACG):** 1: Enables the AFE clock gating 0: Disables the AFE clock gating This bit is only valid when internal TXCLKEN policy is selected (SELTXCLKEN = 0b).
25	0h RW	**ENHSCLK2PORT (ENHSCLK2PORT):** This per-port pin will generate HS clock data pattern for testing purposes for that port and ignore the core data during Tx: 1: The port will transmit 10101..clock data pattern when txen=1. 0: The port will resume to c73usb280_normal operation where data is generated by the controller.
24	0h RW	**SELSBRXBIASEN (SELSBRXBIASEN):** This is to select between the UTMI internal vs. side-band control over the AFE RX bias enable signal. 1 - select side-band control from Intel SIP controller. 0 - select UTMI internally generated control. This is to be used with third party or external controller IP.

Bit Range	Default & Access	Field Name (ID): Description
23	0h RW	**SELSBTXBIASEN (SELSBTXBIASEN):** This is to select between the UTMI internal vs. side-band control over the AFE TX bias enable signal. 1 - select side-band control from Intel SIP controller. 0 - select UTMI internally generated control. This is to be used with third party or external controller IP.
22	0h WO	**RSVD (RSVD):** RSVD
21	0h RW	**SELTXCLKEN (SELTXCLKEN):** This is to select between internal mode vs. side-band control on the TX clock enable signal. 0 - override for TXCLKEN to be internally controlled. To be used with third party or external controller IP. 1 - select side-band control from Intel SIP controller.
20	0h RW	**SELRXCLKEN (SELRXCLKEN):** This is to select between internal mode vs. side-band control on the RX clock enable signal. 0 - override for RXCLKEN to be internally controlled. To be used with third party or external controller IP. 1 - select side-band control from Intel SIP controller.
19:17	2h RW	**PerPort HS Receiver Bias (PERPORTRXISET):** Config bit (per port) HS Receiver Bias current offset bit (2:0) 000 - 80uA 001 - 90uA 010 (default) - 100uA 011 - 110uA 100 - 120uA 101 - 130uA 110 - 140uA 111 - 150uA
16	1h RW	**PerPort Common Mode Pre-charge (PERPORTTXPRECHARGEEN):** Config bit (per port) To select whether to enable or disable the common mode pre-charge during SOP and EOP. Pre-charge hold for 4 cycles before first data transmission. This is to allow common choke ring back to settle down before driving first data bit to meet eye opening spec. 1 - pre-charge ON 0 - pre-charge OFF
15	1h RW	**RX ERROR FIX ENABLE (RXERROR_FIXEN):** When reg_utmi_rxerror_fixen = 1'b1, Rxerror fix is enabled When reg_utmi_rxerror_fixen = 1'b0, Rxerror fix is disabled
14	1h RW	**PerPort Half Bit Pre-emphasis (PERPORTTXPEHALF):** Config bit (per port) to select between half-bit or full-bit implementation 1 - select half-bit pre-emphasis 0 - select full-bit pre-emphasis
13:11	3h RW	**PerPort HS Pre-emphasis Bias (PERPORTPETXISET):** Config bit (per port) HS Pre-emphasis Bias current offset bit2 (2:0) Value Preempen: deempen = 01 Preempen: deempen = 10 Preempen: deempen = 11 (preemphasis strength/demphasis strength) Value (De-emphasis ON) (Pre-emphasis ON) (preemphasis strength/demphasis strength) 000 0mV 0mV 0mV 001 40.5mV 20.5mV 20.5mV/20.5mV 010 60.5mV 30.5mV 30.5mV/30.5mV 011 102mV 52mV 52mV/52mV 100 102mV 52mV 52mV/52mV 101 142mV 72.5mV 72.5mV/72.5mV 110 162.5mV 85mV 85mV/85mV 111 202.5mV 105mV 105mV/105mV

Bit Range	Default & Access	Field Name (ID): Description
10:8	2h RW	**PerPort HS Transmitter Bias (PERPORTTXISET):** Config bit (per port) HS Pre-emphasis Bias current offset bit2 (2:0) Value Preempen: deempen = 01 Preempen: deempen = 10 Preempen: deempen = 11 (preemphasis strength/demphasis strength) Value (De-emphasis ON) (Pre-emphasis ON) (preemphasis strength/demphasis strength) 000 0mV 0mV 0mV 001 40.5mV 20.5mV 20.5mV/20.5mV 010 60.5mV 30.5mV 30.5mV/30.5mV 011 102mV 52mV 52mV/52mV 100 102mV 52mV 52mV/52mV 101 142mV 72.5mV 72.5mV/72.5mV 110 162.5mV 85mV 85mV/85mV 111 202.5mV 105mV 105mV/105mV
7:4	8h RW	**PORTRESERVED_0123 (PORTRESERVED):** PORTRESERVED (PORTRESERVED): Reserved pin for future functionality of the respective ports iusbportreserved[0] is used for differential disconnect purpose. 0 - Differential disconnect and HS squelch uses the same voltage reference (sqref-sqrefb = diffdiscref - diffdiscrefb) 1 - Differential disconnect and HS squelch uses different voltage reference (sqref-sqrefb != diffdiscref - diffdiscrefb) iusbportreserved[1] - iusbafedaten_h_dfxoveride 0 - functional mode. iusbafedaten will be controlled functionally by PCS logic 1 - test mode. The iusbafedaten will be forced to 1 to ensure CL TX is never blocked. iusbportreserved[3:2] - reserved for future functionality.
3	0h RW	**CL TX EMI Reduction Circuit Bypass (CLTXEMIREDCCTBYPASS)** Assert 1 to bypass the EMI reduction circuit. Default is 0b. AFE pin remains as PORTRESERVED for this project.
2	0h RW	**PERPORT ALLPORTZ (PERPORTALLPORTZ):** Legacy pin. No longer carries any functionality as the pull-down enable is now controlled by ilane5:0usbpdnen_h
1	0h RW	**PERPORT PPDPDMBSEL (PERPORTPPDPDMBSEL):** Config bit to select comparator 2:1 mux input either dp or dm for HS TX per port current calibration 1 - select dp (for debug purpose only) 0 - select dm (without 1.5K pullup)
0	1h RW	**PERPORT TX EOS protection enable (PERPORTTXEOS):** Set to 1b enables the TX EOS protection. To protect high speed driver passgate.

33.2 c73usb280_USB2 PER PORT 2 (USB2_PER_PORT_2_PP0)—Offset 4126h

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 829248h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO/V	**RSVD (RSVD):** RSVD

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**Manual Override of Device Idle Pullup Enable Select (REG_DEVIDLE_SEL):** Feed directly to SHIP or AFE : SHIP Functional or Test Mode : Functional 1: To enable manual override on device Idle pullup enable. Enabling/Disabling come from reg_didle_force_onoff. 0: Disable Manual override on device idle pullup enable. Device idle pullup enable based on functional logic.
26	0h RW	**Device Idle Force On Off (REG_DEVIDLE):** Feed directly to SHIP or AFE : SHIP Functional or Test Mode : Functional 1: To force turn on of device idle pullup enable. 0: To force turn off of device idle pullup enable.
25	0h RW	**HSSKEWSEL (HSSKEWSEL):** Select the skew direction for HS transition. Hsskewsel | Skew direction 0 | Delay HS Driver 0)1 Transition (default) 1 | Delay HS Driver 1)0 Transition
24:23	1h RW	**Per Port HS Transmitter Emphasis (IUSBTXEMPHASISEN):** Config bit (per port) Enables HS transmitter Emphasis for the respective ports Value Description 00 Emphasis OFF 01 De-emphasis ON (strength = TXISET+PETXISET) Default 10 Pre-emphasis ON (strength = TXISET+PETXISET) 11 Pre-emphasis and De-emphasis ON (strength = TXISET+PETXISET)
22	0h RW	**ADPLL clock monitoring through TX (IUSBENVCOCLKMON):** Signal to enable ADPLL clock monitoring through TX driver.
21	0h RW	**DISABLE SOF WINDOW FOR DISCONNECT (DISABLE_SOFWINDOW_CKBIT):** 1: Enable sampling only during SOF window for disconnect. 0:Sampling not limited to SOF window.
20	0h RW	**RSVD (RSVD_1)**
19	0h RW	**EOS Shunt always on (SHUNT_AON):** To force shunt FSM to always on and not run shunt detect FSM.
18	0h RW	**EOS Shunt legcheck (SHUNTLEGCHK):** To Enable EOS Shunt protection legcheck.
17:15	5h RW	**EOS Shunt configuration (IUSBCFGSHUNT):** Set Shunt strength control trim during VIHz 000 - Turn off shunt 100 - Lowest Shunt strength 101 - Low Shunt Strength 110 - Medium Shunt Strength 111 - High Shunt Strength
14	0h RW	**VIHz EOS Shunt Protection Off Control (VESPONC):** 1: Force the shunt to turn statically OFF. 0: Dynamic shunt ON/OFF mode based on port suspend. Bits (VESPOFFC, VESPONC) should never be set to 11b.
13	0h RW	**VIHz EOS Shunt Protection On Control (VESPOFFC):** 1: Force the shunt to turn statically ON. 0: Dynamic shunt ON/OFF mode based on port suspend. Bits (VESPOFFC, VESPONC) should never be set to 11b.
12:11	2h RW	**CFGHYSCTL (CFGHYSCTL):** Hysteresis control for Classic Single Ended Rx 00 - 0mV 01 - 25mV 10 - 75mV (default) 11 - 125mV

Bit Range	Default & Access	Field Name (ID): Description
10:9	1h RW	**CFGBIASCTL (CFGBIASCTL):** Bias control for Classic Differential Rx 00 - 50uA 01 - 60uA (default) 1X - 70uA
8	0h RW	**DISSLEWCTL (DISSLEWCTL):** DISSLEWCTL (DISSLEWCTL): Feed directly to SHIP or AFE : AFE Functional or Test Mode : Functional 0 - enable slew rate control during Classic Tx (default) 1 - disable slew rate control during Classic Tx
7:5	2h RW	**CLSLEWCTLPD (CLSLEWCTLPD):** Pull-Down Slew rate control for Classic Low Speed Tx 000 - slowest slew rate 001 - slow slew rate 010 - c73usb280_Normal (default) 100 - fast slew rate
4:2	2h RW	**CLSLEWCTLPU (CLSLEWCTLPU):** Slew rate control for Classic Low Speed Tx 000 - slowest slew rate 001 - slow slew rate 010 - c73usb280_Normal (default) 100 - fast slew rate
1:0	0h RW	**HSNPREDRVSEL (HSNPREDRVSEL):** Delay/skews strength Control for HS driver. Use together with HSSKEWSEL. npredrvsel(1) | npredrvsel(0) | Skew Offset 0 | 0 | No Skew (default) 0 | 1 | 30mV 1 | 0 | 40mV 1 | 1 | 50mV

33.3 c73usb280_USB2 PER PORT (USB2_PER_PORT_PP1)—Offset 4200h

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 5DA81h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Per-Port UTMI_CLK Output Clock Gating Enable (PPUOCGE):** 1: Dynamic Clock Gating is enabled based on L1/L2 suspend indicator. This is the recommended setting for SYNOPSYS based controllers. 0: Output UTMI_CLK is running unless USB PLL is shutdown. This is the recommended setting for FRESCO based controllers.
30	0h RW	**RX EN AFE Control (RXENAC):** 1: Enables the dynamic RX EN behavior. This allows compensation code updates at various boundaries. 0: Disables the dynamic RX EN behavior (function of TXVALID only). RXENAC, RXBIASAC: 00: Legacy mode 01: Illegal 10: Allows additional code update window 11: Allows additional code update window with power saving
29	0h RW	**RX BIAS AFE Control (RXBIASAC):** 1: Enables the AFE bias control 0: Disables the AFE bias control This bit is only valid when internal RXBIAS policy is selected (SELSBRXBIASEN = 0b).

Bit Range	Default & Access	Field Name (ID): Description
28	0h RW	**TX BIAS AFE Control (TXBIASAC):** 1: Enables the AFE bias control 0: Disables the AFE bias control This bit is only valid when internal TXBIAS policy is selected (SELSBTXBIASEN = 0b).
27	0h RW	**RX CLKEN AFE Clock Gating (RXCLKENACG):** 1: Enables the AFE clock gating 0: Disables the AFE clock gating This bit is only valid when internal RXCLKEN policy is selected (SELRXCLKEN = 0b).
26	0h RW	**TX CLKEN AFE Clock Gating (TXCLKENACG):** 1: Enables the AFE clock gating 0: Disables the AFE clock gating This bit is only valid when internal TXCLKEN policy is selected (SELTXCLKEN = 0b).
25	0h RW	**ENHSCLK2PORT (ENHSCLK2PORT):** This per-port pin will generate HS clock data pattern for testing purposes for that port and ignore the core data during Tx: 1: The port will transmit 10101..clock data pattern when txen=1. 0: The port will resume to c73usb280_normal operation where data is generated by the controller.
24	0h RW	**SELSBRXBIASEN (SELSBRXBIASEN):** This is to select between the UTMI internal vs. side-band control over the AFE RX bias enable signal. 1 - select side-band control from Intel SIP controller. 0 - select UTMI internally generated control. This is to be used with third party or external controller IP.
23	0h RW	**SELSBTXBIASEN (SELSBTXBIASEN):** This is to select between the UTMI internal vs. side-band control over the AFE TX bias enable signal. 1 - select side-band control from Intel SIP controller. 0 - select UTMI internally generated control. This is to be used with third party or external controller IP.
22	0h WO	**RSVD (RSVD):** RSVD
21	0h RW	**SELTXCLKEN (SELTXCLKEN):** This is to select between internal mode vs. side-band control on the TX clock enable signal. 0 - override for TXCLKEN to be internally controlled. To be used with third party or external controller IP. 1 - select side-band control from Intel SIP controller.
20	0h RW	**SELRXCLKEN (SELRXCLKEN):** This is to select between internal mode vs. side-band control on the RX clock enable signal. 0 - override for RXCLKEN to be internally controlled. To be used with third party or external controller IP. 1 - select side-band control from Intel SIP controller.
19:17	2h RW	**PerPort HS Receiver Bias (PERPORTRXISET):** Config bit (per port) HS Receiver Bias current offset bit (2:0) 000 - 80uA 001 - 90uA 010 (default) - 100uA 011 - 110uA 100 - 120uA 101 - 130uA 110 - 140uA 111 - 150uA
16	1h RW	**PerPort Common Mode Pre-charge (PERPORTTXPRECHARGEEN):** Config bit (per port) To select whether to enable or disable the common mode pre-charge during SOP and EOP. Pre-charge hold for 4 cycles before first data transmission. This is to allow common choke ring back to settle down before driving first data bit to meet eye opening spec. 1 - pre-charge ON 0 - pre-charge OFF
15	1h RW	**RX ERROR FIX ENABLE (RXERROR_FIXEN):** When reg_utmi_rxerror_fixen = 1'b1, Rxerror fix is enabled When reg_utmi_rxerror_fixen = 1'b0, Rxerror fix is disabled

Bit Range	Default & Access	Field Name (ID): Description
14	1h RW	**PerPort Half Bit Pre-emphasis (PERPORTTXPEHALF):** Config bit (per port) to select between half-bit or full-bit implementation 1 - select half-bit pre-emphasis 0 - select full-bit pre-emphasis
13:11	3h RW	**PerPort HS Pre-emphasis Bias (PERPORTPETXISET):** Config bit (per port) HS Pre-emphasis Bias current offset bit2 (2:0) Value Preempen: deempen = 01 Preempen: deempen = 10 Preempen: deempen = 11 (preemphasis strength/demphasis strength) Value (De-emphasis ON) (Pre-emphasis ON) (preemphasis strength/demphasis strength) 000 0mV 0mV 0mV 001 40.5mV 20.5mV 20.5mV/20.5mV 010 60.5mV 30.5mV 30.5mV/30.5mV 011 102mV 52mV 52mV/52mV 100 102mV 52mV 52mV/52mV 101 142mV 72.5mV 72.5mV/72.5mV 110 162.5mV 85mV 85mV/85mV 111 202.5mV 105mV 105mV/105mV
10:8	2h RW	**PerPort HS Transmitter Bias (PERPORTTXISET):** Config bit (per port) HS Pre-emphasis Bias current offset bit2 (2:0) Value Preempen: deempen = 01 Preempen: deempen = 10 Preempen: deempen = 11 (preemphasis strength/demphasis strength) Value (De-emphasis ON) (Pre-emphasis ON) (preemphasis strength/demphasis strength) 000 0mV 0mV 0mV 001 40.5mV 20.5mV 20.5mV/20.5mV 010 60.5mV 30.5mV 30.5mV/30.5mV 011 102mV 52mV 52mV/52mV 100 102mV 52mV 52mV/52mV 101 142mV 72.5mV 72.5mV/72.5mV 110 162.5mV 85mV 85mV/85mV 111 202.5mV 105mV 105mV/105mV
7:4	8h RW	**PORTRESERVED_0123 (PORTRESERVED):** PORTRESERVED (PORTRESERVED): Reserved pin for future functionality of the respective ports iusbportreserved[0] is used for differential disconnect purpose. 0 - Differential disconnect and HS squelch uses the same voltage reference (sqref-sqrefb = diffdiscref - diffdiscrefb) 1 - Differential disconnect and HS squelch uses different voltage reference (sqref-sqrefb != diffdiscref - diffdiscrefb) iusbportreserved[1] - iusbafedaten_h_dfxoveride 0 - functional mode. iusbafedaten will be controlled functionally by PCS logic 1 - test mode. The iusbafedaten will be forced to 1 to ensure CL TX is never blocked. iusbportreserved[3:2] - reserved for future functionality.
3	0h RW	**CL TX EMI Reduction Circuit Bypass (CLTXEMIREDCCTBYPASS):** Assert 1 to bypass the EMI reduction circuit. Default is 0b. AFE pin remains as PORTRESERVED for this project.
2	0h RW	**PERPORT ALLPORTZ (PERPORTALLPORTZ):** Legacy pin. No longer carries any functionality as the pull-down enable is now controlled by ilane5:0usbpdnen_h
1	0h RW	**PERPORT PPDPDMBSEL (PERPORTPPDPDMBSEL):** Config bit to select comparator 2:1 mux input either dp or dm for HS TX per port current calibration 1 - select dp (for debug purpose only) 0 - select dm (without 1.5K pullup)
0	1h RW	**PERPORT TX EOS protection enable (PERPORTTXEOS):** Set to 1b enables the TX EOS protection. To protect high speed driver passgate.

33.4 c73usb280_USB2 PER PORT 2 (USB2_PER_PORT_2_PP1)—Offset 4226h

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 829248h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO/V	**RSVD (RSVD):** RSVD
27	0h RW	**Manual Override of Device Idle Pullup Enable Select (REG_DEVIDLE_SEL):** Feed directly to SHIP or AFE : SHIP Functional or Test Mode : Functional 1: To enable manual override on device Idle pullup enable. Enabling/Disabling come from reg_didle_force_onoff. 0: Disable Manual override on device idle pullup enable. Device idle pullup enable based on functional logic.
26	0h RW	**Device Idle Force On Off (REG_DEVIDLE):** Feed directly to SHIP or AFE : SHIP Functional or Test Mode : Functional 1: To force turn on of device idle pullup enable. 0: To force turn off of device idle pullup enable.
25	0h RW	**HSSKEWSEL (HSSKEWSEL):** Select the skew direction for HS transition. Hsskewsel | Skew direction 0 | Delay HS Driver 0)1 Transition (default) 1 | Delay HS Driver 1)0 Transition
24:23	1h RW	**Per Port HS Transmitter Emphasis (IUSBTXEMPHASISEN):** Config bit (per port) Enables HS transmitter Emphasis for the respective ports Value Description 00 Emphasis OFF 01 De-emphasis ON (strength = TXISET+PETXISET) Default 10 Pre-emphasis ON (strength = TXISET+PETXISET) 11 Pre-emphasis and De-emphasis ON (strength = TXISET+PETXISET)
22	0h RW	**ADPLL clock monitoring through TX (IUSBENVCOCLKMON):** Signal to enable ADPLL clock monitoring through TX driver.
21	0h RW	**DISABLE SOF WINDOW FOR DISCONNECT (DISABLE_SOFWINDOW_CKBIT):** 1: Enable sampling only during SOF window for disconnect. 0:Sampling not limited to SOF window.
20	0h RW	**RSVD (RSVD_1)**
19	0h RW	**EOS Shunt always on (SHUNT_AON):** To force shunt FSM to always on and not run shunt detect FSM.
18	0h RW	**EOS Shunt legcheck (SHUNTLEGCHK):** To Enable EOS Shunt protection legcheck.
17:15	5h RW	**EOS Shunt configuration (IUSBCFGSHUNT):** Set Shunt strength control trim during VIHz 000 - Turn off shunt 100 - Lowest Shunt strength 101 - Low Shunt Strength 110 - Medium Shunt Strength 111 - High Shunt Strength
14	0h RW	**VIHz EOS Shunt Protection Off Control (VESPONC):** 1: Force the shunt to turn statically OFF. 0: Dynamic shunt ON/OFF mode based on port suspend. Bits (VESPOFFC, VESPONC) should never be set to 11b.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**VIHz EOS Shunt Protection On Control (VESPOFFC):** 1: Force the shunt to turn statically ON. 0: Dynamic shunt ON/OFF mode based on port suspend. Bits (VESPOFFC, VESPONC) should never be set to 11b.
12:11	2h RW	**CFGHYSCTL (CFGHYSCTL):** Hysteresis control for Classic Single Ended Rx 00 - 0mV 01 - 25mV 10 - 75mV (default) 11 - 125mV
10:9	1h RW	**CFGBIASCTL (CFGBIASCTL):** Bias control for Classic Differential Rx 00 - 50uA 01 - 60uA (default) 1X - 70uA
8	0h RW	**DISSLEWCTL (DISSLEWCTL):** DISSLEWCTL (DISSLEWCTL): Feed directly to SHIP or AFE : AFE Functional or Test Mode : Functional 0 - enable slew rate control during Classic Tx (default) 1 - disable slew rate control during Classic Tx
7:5	2h RW	**CLSLEWCTLPD (CLSLEWCTLPD):** Pull-Down Slew rate control for Classic Low Speed Tx 000 - slowest slew rate 001 - slow slew rate 010 - c73usb280_Normal (default) 100 - fast slew rate
4:2	2h RW	**CLSLEWCTLPU (CLSLEWCTLPU):** Slew rate control for Classic Low Speed Tx 000 - slowest slew rate 001 - slow slew rate 010 - c73usb280_Normal (default) 100 - fast slew rate
1:0	0h RW	**HSNPREDRVSEL (HSNPREDRVSEL):** Delay/skews strength Control for HS driver. Use together with HSSKEWSEL. npredrvsel(1) | npredrvsel(0) | Skew Offset 0 | 0 | No Skew (default) 0 | 1 | 30mV 1 | 0 | 40mV 1 | 1 | 50mV

33.5 c73usb280_USB2 PER PORT (USB2_PER_PORT_PP2)—Offset 4300h

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 5DA81h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Per-Port UTMI_CLK Output Clock Gating Enable (PPUOCGE):** 1: Dynamic Clock Gating is enabled based on L1/L2 suspend indicator. This is the recommended setting for SYNOPSYS based controllers. 0: Output UTMI_CLK is running unless USB PLL is shutdown. This is the recommended setting for FRESCO based controllers.

Bit Range	Default & Access	Field Name (ID): Description
30	0h RW	**RX EN AFE Control (RXENAC):** 1: Enables the dynamic RX EN behavior. This allows compensation code updates at various boundaries. 0: Disables the dynamic RX EN behavior (function of TXVALID only). RXENAC, RXBIASAC: 00: Legacy mode 01: Illegal 10: Allows additional code update window 11: Allows additional code update window with power saving
29	0h RW	**RX BIAS AFE Control (RXBIASAC):** 1: Enables the AFE bias control 0: Disables the AFE bias control This bit is only valid when internal RXBIAS policy is selected (SELSBRXBIASEN = 0b).
28	0h RW	**TX BIAS AFE Control (TXBIASAC):** 1: Enables the AFE bias control 0: Disables the AFE bias control This bit is only valid when internal TXBIAS policy is selected (SELSBTXBIASEN = 0b).
27	0h RW	**RX CLKEN AFE Clock Gating (RXCLKENACG):** 1: Enables the AFE clock gating 0: Disables the AFE clock gating This bit is only valid when internal RXCLKEN policy is selected (SELRXCLKEN = 0b).
26	0h RW	**TX CLKEN AFE Clock Gating (TXCLKENACG):** 1: Enables the AFE clock gating 0: Disables the AFE clock gating This bit is only valid when internal TXCLKEN policy is selected (SELTXCLKEN = 0b).
25	0h RW	**ENHSCLK2PORT (ENHSCLK2PORT):** This per-port pin will generate HS clock data pattern for testing purposes for that port and ignore the core data during Tx: 1: The port will transmit 10101..clock data pattern when txen=1. 0: The port will resume to c73usb280_normal operation where data is generated by the controller.
24	0h RW	**SELSBRXBIASEN (SELSBRXBIASEN):** This is to select between the UTMI internal vs. side-band control over the AFE RX bias enable signal. 1 - select side-band control from Intel SIP controller. 0 - select UTMI internally generated control. This is to be used with third party or external controller IP.
23	0h RW	**SELSBTXBIASEN (SELSBTXBIASEN):** This is to select between the UTMI internal vs. side-band control over the AFE TX bias enable signal. 1 - select side-band control from Intel SIP controller. 0 - select UTMI internally generated control. This is to be used with third party or external controller IP.
22	0h WO	**RSVD (RSVD):** RSVD
21	0h RW	**SELTXCLKEN (SELTXCLKEN):** This is to select between internal mode vs. side-band control on the TX clock enable signal. 0 - override for TXCLKEN to be internally controlled. To be used with third party or external controller IP. 1 - select side-band control from Intel SIP controller.
20	0h RW	**SELRXCLKEN (SELRXCLKEN):** This is to select between internal mode vs. side-band control on the RX clock enable signal. 0 - override for RXCLKEN to be internally controlled. To be used with third party or external controller IP. 1 - select side-band control from Intel SIP controller.

Bit Range	Default & Access	Field Name (ID): Description
19:17	2h RW	**PerPort HS Receiver Bias (PERPORTRXISET):** Config bit (per port) HS Receiver Bias current offset bit (2:0) 000 - 80uA 001 - 90uA 010 (default) - 100uA 011 - 110uA 100 - 120uA 101 - 130uA 110 - 140uA 111 - 150uA
16	1h RW	**PerPort Common Mode Pre-charge (PERPORTTXPRECHARGEEN):** Config bit (per port) To select whether to enable or disable the common mode pre-charge during SOP and EOP. Pre-charge hold for 4 cycles before first data transmission. This is to allow common choke ring back to settle down before driving first data bit to meet eye opening spec. 1 - pre-charge ON 0 - pre-charge OFF
15	1h RW	**RX ERROR FIX ENABLE (RXERROR_FIXEN):** When reg_utmi_rxerror_fixen = 1'b1, Rxerror fix is enabled When reg_utmi_rxerror_fixen = 1'b0, Rxerror fix is disabled
14	1h RW	**PerPort Half Bit Pre-emphasis (PERPORTTXPEHALF):** Config bit (per port) to select between half-bit or full-bit implementation 1 - select half-bit pre-emphasis 0 - select full-bit pre-emphasis
13:11	3h RW	**PerPort HS Pre-emphasis Bias (PERPORTPETXISET):** Config bit (per port) HS Pre-emphasis Bias current offset bit2 (2:0) Value Preempen: deempen = 01 Preempen: deempen = 10 Preempen: deempen = 11 (preemphasis strength/demphasis strength) Value (De-emphasis ON) (Pre-emphasis ON) (preemphasis strength/demphasis strength) 000 0mV 0mV 0mV 001 40.5mV 20.5mV 20.5mV/20.5mV 010 60.5mV 30.5mV 30.5mV/30.5mV 011 102mV 52mV 52mV/52mV 100 102mV 52mV 52mV/52mV 101 142mV 72.5mV 72.5mV/72.5mV 110 162.5mV 85mV 85mV/85mV 111 202.5mV 105mV 105mV/105mV
10:8	2h RW	**PerPort HS Transmitter Bias (PERPORTTXISET):** Config bit (per port) HS Pre-emphasis Bias current offset bit2 (2:0) Value Preempen: deempen = 01 Preempen: deempen = 10 Preempen: deempen = 11 (preemphasis strength/demphasis strength) Value (De-emphasis ON) (Pre-emphasis ON) (preemphasis strength/demphasis strength) 000 0mV 0mV 0mV 001 40.5mV 20.5mV 20.5mV/20.5mV 010 60.5mV 30.5mV 30.5mV/30.5mV 011 102mV 52mV 52mV/52mV 100 102mV 52mV 52mV/52mV 101 142mV 72.5mV 72.5mV/72.5mV 110 162.5mV 85mV 85mV/85mV 111 202.5mV 105mV 105mV/105mV
7:4	8h RW	**PORTRESERVED_0123 (PORTRESERVED):** PORTRESERVED (PORTRESERVED): Reserved pin for future functionality of the respective ports iusbportreserved[0] is used for differential disconnect purpose. 0 - Differential disconnect and HS squelch uses the same voltage reference (sqref-sqrefb = diffdiscref - diffdiscrefb) 1 - Differential disconnect and HS squelch uses different voltage reference (sqref-sqrefb != diffdiscref - diffdiscrefb) iusbportreserved[1] - iusbafedaten_h_dfxoveride 0 - functional mode. iusbafedaten will be controlled functionally by PCS logic 1 - test mode. The iusbafedaten will be forced to 1 to ensure CL TX is never blocked. iusbportreserved[3:2] - reserved for future functionality.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**CL TX EMI Reduction Circuit Bypass (CLTXEMIREDCCTBYPASS):** Assert 1 to bypass the EMI reduction circuit. Default is 0b. AFE pin remains as PORTRESERVED for this project.
2	0h RW	**PERPORT ALLPORTZ (PERPORTALLPORTZ):** Legacy pin. No longer carries any functionality as the pull-down enable is now controlled by ilane5:0usbpdnen_h
1	0h RW	**PERPORT PPDPDMBSEL (PERPORTPPDPDMBSEL):** Config bit to select comparator 2:1 mux input either dp or dm for HS TX per port current calibration 1 - select dp (for debug purpose only) 0 - select dm (without 1.5K pullup)
0	1h RW	**PERPORT TX EOS protection enable (PERPORTTXEOS):** Set to 1b enables the TX EOS protection. To protect high speed driver passgate.

33.6 c73usb280_USB2 PER PORT 2 (USB2_PER_PORT_2_PP2)—Offset 4326h

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 829248h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO/V	**RSVD (RSVD):** RSVD
27	0h RW	**Manual Override of Device Idle Pullup Enable Select (REG_DEVIDLE_SEL):** Feed directly to SHIP or AFE : SHIP Functional or Test Mode : Functional 1: To enable manual override on device Idle pullup enable. Enabling/Disabling come from reg_didle_force_onoff. 0: Disable Manual override on device idle pullup enable. Device idle pullup enable based on functional logic.
26	0h RW	**Device Idle Force On Off (REG_DEVIDLE):** Feed directly to SHIP or AFE : SHIP Functional or Test Mode : Functional 1: To force turn on of device idle pullup enable. 0: To force turn off of device idle pullup enable.
25	0h RW	**HSSKEWSEL (HSSKEWSEL):** Select the skew direction for HS transition. Hsskewsel | Skew direction 0 | Delay HS Driver 0)1 Transition (default) 1 | Delay HS Driver 1)0 Transition
24:23	1h RW	**Per Port HS Transmitter Emphasis (IUSBTXEMPHASISEN):** Config bit (per port) Enables HS transmitter Emphasis for the respective ports Value Description 00 Emphasis OFF 01 De-emphasis ON (strength = TXISET+PETXISET) Default 10 Pre-emphasis ON (strength = TXISET+PETXISET) 11 Pre-emphasis and De-emphasis ON (strength = TXISET+PETXISET)
22	0h RW	**ADPLL clock monitoring through TX (IUSBENVCOCLKMON):** Signal to enable ADPLL clock monitoring through TX driver.
21	0h RW	**DISABLE SOF WINDOW FOR DISCONNECT (DISABLE_SOFWINDOW_CKBIT):** 1: Enable sampling only during SOF window for disconnect. 0:Sampling not limited to SOF window.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**RSVD (RSVD_1)**
19	0h RW	**EOS Shunt always on (SHUNT_AON):** To force shunt FSM to always on and not run shunt detect FSM.
18	0h RW	**EOS Shunt legcheck (SHUNTLEGCHK):** To Enable EOS Shunt protection legcheck.
17:15	5h RW	**EOS Shunt configuration (IUSBCFGSHUNT):** Set Shunt strength control trim during VIHz 000 - Turn off shunt 100 - Lowest Shunt strength 101 - Low Shunt Strength 110 - Medium Shunt Strength 111 - High Shunt Strength
14	0h RW	**VIHz EOS Shunt Protection Off Control (VESPONC):** 1: Force the shunt to turn statically OFF. 0: Dynamic shunt ON/OFF mode based on port suspend. Bits (VESPOFFC, VESPONC) should never be set to 11b.
13	0h RW	**VIHz EOS Shunt Protection On Control (VESPOFFC):** 1: Force the shunt to turn statically ON. 0: Dynamic shunt ON/OFF mode based on port suspend. Bits (VESPOFFC, VESPONC) should never be set to 11b.
12:11	2h RW	**CFGHYSCTL (CFGHYSCTL):** Hysteresis control for Classic Single Ended Rx 00 - 0mV 01 - 25mV 10 - 75mV (default) 11 - 125mV
10:9	1h RW	**CFGBIASCTL (CFGBIASCTL):** Bias control for Classic Differential Rx 00 - 50uA 01 - 60uA (default) 1X - 70uA
8	0h RW	**DISSLEWCTL (DISSLEWCTL):** DISSLEWCTL (DISSLEWCTL): Feed directly to SHIP or AFE : AFE Functional or Test Mode : Functional 0 - enable slew rate control during Classic Tx (default) 1 - disable slew rate control during Classic Tx
7:5	2h RW	**CLSLEWCTLPD (CLSLEWCTLPD):** Pull-Down Slew rate control for Classic Low Speed Tx 000 - slowest slew rate 001 - slow slew rate 010 - c73usb280_Normal (default) 100 - fast slew rate
4:2	2h RW	**CLSLEWCTLPU (CLSLEWCTLPU):** Slew rate control for Classic Low Speed Tx 000 - slowest slew rate 001 - slow slew rate 010 - c73usb280_Normal (default) 100 - fast slew rate
1:0	0h RW	**HSNPREDRVSEL (HSNPREDRVSEL):** Delay/skews strength Control for HS driver. Use together with HSSKEWSEL. npredrvsel(1) | npredrvsel(0) | Skew Offset 0 | 0 | No Skew (default) 0 | 1 | 30mV 1 | 0 | 40mV 1 | 1 | 50mV

33.7 c73usb280_USB2 PER PORT (USB2_PER_PORT_PP3)—Offset 4400h

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 5DA81h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Per-Port UTMI_CLK Output Clock Gating Enable (PPUOCGE):** 1: Dynamic Clock Gating is enabled based on L1/L2 suspend indicator. This is the recommended setting for SYNOPSYS based controllers. 0: Output UTMI_CLK is running unless USB PLL is shutdown. This is the recommended setting for FRESCO based controllers.
30	0h RW	**RX EN AFE Control (RXENAC):** 1: Enables the dynamic RX EN behavior. This allows compensation code updates at various boundaries. 0: Disables the dynamic RX EN behavior (function of TXVALID only). RXENAC, RXBIASAC: 00: Legacy mode 01: Illegal 10: Allows additional code update window 11: Allows additional code update window with power saving
29	0h RW	**RX BIAS AFE Control (RXBIASAC):** 1: Enables the AFE bias control 0: Disables the AFE bias control This bit is only valid when internal RXBIAS policy is selected (SELSBRXBIASEN = 0b).
28	0h RW	**TX BIAS AFE Control (TXBIASAC):** 1: Enables the AFE bias control 0: Disables the AFE bias control This bit is only valid when internal TXBIAS policy is selected (SELSBTXBIASEN = 0b).
27	0h RW	**RX CLKEN AFE Clock Gating (RXCLKENACG):** 1: Enables the AFE clock gating 0: Disables the AFE clock gating This bit is only valid when internal RXCLKEN policy is selected (SELRXCLKEN = 0b).
26	0h RW	**TX CLKEN AFE Clock Gating (TXCLKENACG):** 1: Enables the AFE clock gating 0: Disables the AFE clock gating This bit is only valid when internal TXCLKEN policy is selected (SELTXCLKEN = 0b).
25	0h RW	**ENHSCLK2PORT (ENHSCLK2PORT):** This per-port pin will generate HS clock data pattern for testing purposes for that port and ignore the core data during Tx: 1: The port will transmit 10101..clock data pattern when txen=1. 0: The port will resume to c73usb280_normal operation where data is generated by the controller.
24	0h RW	**SELSBRXBIASEN (SELSBRXBIASEN):** This is to select between the UTMI internal vs. side-band control over the AFE RX bias enable signal. 1 - select side-band control from Intel SIP controller. 0 - select UTMI internally generated control. This is to be used with third party or external controller IP.
23	0h RW	**SELSBTXBIASEN (SELSBTXBIASEN):** This is to select between the UTMI internal vs. side-band control over the AFE TX bias enable signal. 1 - select side-band control from Intel SIP controller. 0 - select UTMI internally generated control. This is to be used with third party or external controller IP.
22	0h WO	**RSVD (RSVD):** RSVD

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**SELTXCLKEN (SELTXCLKEN):** This is to select between internal mode vs. side-band control on the TX clock enable signal. 0 - override for TXCLKEN to be internally controlled. To be used with third party or external controller IP. 1 - select side-band control from Intel SIP controller.
20	0h RW	**SELRXCLKEN (SELRXCLKEN):** This is to select between internal mode vs. side-band control on the RX clock enable signal. 0 - override for RXCLKEN to be internally controlled. To be used with third party or external controller IP. 1 - select side-band control from Intel SIP controller.
19:17	2h RW	**PerPort HS Receiver Bias (PERPORTRXISET):** Config bit (per port) HS Receiver Bias current offset bit (2:0) 000 - 80uA 001 - 90uA 010 (default) - 100uA 011 - 110uA 100 - 120uA 101 - 130uA 110 - 140uA 111 - 150uA
16	1h RW	**PerPort Common Mode Pre-charge (PERPORTTXPRECHARGEEN):** Config bit (per port) To select whether to enable or disable the common mode pre-charge during SOP and EOP. Pre-charge hold for 4 cycles before first data transmission. This is to allow common choke ring back to settle down before driving first data bit to meet eye opening spec. 1 - pre-charge ON 0 - pre-charge OFF
15	1h RW	**RX ERROR FIX ENABLE (RXERROR_FIXEN):** When reg_utmi_rxerror_fixen = 1'b1, Rxerror fix is enabled When reg_utmi_rxerror_fixen = 1'b0, Rxerror fix is disabled
14	1h RW	**PerPort Half Bit Pre-emphasis (PERPORTTXPEHALF):** Config bit (per port) to select between half-bit or full-bit implementation 1 - select half-bit pre-emphasis 0 - select full-bit pre-emphasis
13:11	3h RW	**PerPort HS Pre-emphasis Bias (PERPORTPETXISET):** Config bit (per port) HS Pre-emphasis Bias current offset bit2 (2:0) Value Preempen: deempen = 01 Preempen: deempen = 10 Preempen: deempen = 11 (preemphasis strength/demphasis strength) Value (De-emphasis ON) (Pre-emphasis ON) (preemphasis strength/demphasis strength) 000 0mV 0mV 0mV 001 40.5mV 20.5mV 20.5mV/20.5mV 010 60.5mV 30.5mV 30.5mV/30.5mV 011 102mV 52mV 52mV/52mV 100 102mV 52mV 52mV/52mV 101 142mV 72.5mV 72.5mV/72.5mV 110 162.5mV 85mV 85mV/85mV 111 202.5mV 105mV 105mV/105mV
10:8	2h RW	**PerPort HS Transmitter Bias (PERPORTTXISET):** Config bit (per port) HS Pre-emphasis Bias current offset bit2 (2:0) Value Preempen: deempen = 01 Preempen: deempen = 10 Preempen: deempen = 11 (preemphasis strength/demphasis strength) Value (De-emphasis ON) (Pre-emphasis ON) (preemphasis strength/demphasis strength) 000 0mV 0mV 0mV 001 40.5mV 20.5mV 20.5mV/20.5mV 010 60.5mV 30.5mV 30.5mV/30.5mV 011 102mV 52mV 52mV/52mV 100 102mV 52mV 52mV/52mV 101 142mV 72.5mV 72.5mV/72.5mV 110 162.5mV 85mV 85mV/85mV 111 202.5mV 105mV 105mV/105mV

Bit Range	Default & Access	Field Name (ID): Description
7:4	8h RW	**PORTRESERVED_0123 (PORTRESERVED):** PORTRESERVED (PORTRESERVED): Reserved pin for future functionality of the respective ports iusbportreserved[0] is used for differential disconnect purpose. 0 - Differential disconnect and HS squelch uses the same voltage reference (sqref-sqrefb = diffdiscref - diffdiscrefb) 1 - Differential disconnect and HS squelch uses different voltage reference (sqref-sqrefb != diffdiscref - diffdiscrefb) iusbportreserved[1] - iusbafedaten_h_dfxoveride 0 - functional mode. iusbafedaten will be controlled functionally by PCS logic 1 - test mode. The iusbafedaten will be forced to 1 to ensure CL TX is never blocked. iusbportreserved[3:2] - reserved for future functionality.
3	0h RW	**CL TX EMI Reduction Circuit Bypass (CLTXEMIREDCCTBYPASS):** Assert 1 to bypass the EMI reduction circuit. Default is 0b. AFE pin remains as PORTRESERVED for this project.
2	0h RW	**PERPORT ALLPORTZ (PERPORTALLPORTZ):** Legacy pin. No longer carries any functionality as the pull-down enable is now controlled by ilane5:0usbpdnen_h
1	0h RW	**PERPORT PPDPDMBSEL (PERPORTPPDPDMBSEL):** Config bit to select comparator 2:1 mux input either dp or dm for HS TX per port current calibration 1 - select dp (for debug purpose only) 0 - select dm (without 1.5K pullup)
0	1h RW	**PERPORT TX EOS protection enable (PERPORTTXEOS):** Set to 1b enables the TX EOS protection. To protect high speed driver passgate.

33.8 c73usb280_USB2 PER PORT 2 (USB2_PER_PORT_2_PP3)—Offset 4426h

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 829248h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO/V	**RSVD (RSVD):** RSVD
27	0h RW	**Manual Override of Device Idle Pullup Enable Select (REG_DEVIDLE_SEL):** Feed directly to SHIP or AFE : SHIP Functional or Test Mode : Functional 1: To enable manual override on device Idle pullup enable. Enabling/Disabling come from reg_didle_force_onoff. 0: Disable Manual override on device idle pullup enable. Device idle pullup enable based on functional logic.
26	0h RW	**Device Idle Force On Off (REG_DEVIDLE):** Feed directly to SHIP or AFE : SHIP Functional or Test Mode : Functional 1: To force turn on of device idle pullup enable. 0: To force turn off of device idle pullup enable.
25	0h RW	**HSSKEWSEL (HSSKEWSEL):** Select the skew direction for HS transition. Hsskewsel | Skew direction 0 | Delay HS Driver 0)1 Transition (default) 1 | Delay HS Driver 1)0 Transition

Bit Range	Default & Access	Field Name (ID): Description
24:23	1h RW	**Per Port HS Transmitter Emphasis (IUSBTXEMPHASISEN):** Config bit (per port) Enables HS transmitter Emphasis for the respective ports Value Description 00 Emphasis OFF 01 De-emphasis ON (strength = TXISET+PETXISET) Default 10 Pre-emphasis ON (strength = TXISET+PETXISET) 11 Pre-emphasis and De-emphasis ON (strength = TXISET+PETXISET)
22	0h RW	**ADPLL clock monitoring through TX (IUSBENVCOCLKMON):** Signal to enable ADPLL clock monitoring through TX driver.
21	0h RW	**DISABLE SOF WINDOW FOR DISCONNECT (DISABLE_SOFWINDOW_CKBIT):** 1: Enable sampling only during SOF window for disconnect. 0:Sampling not limited to SOF window.
20	0h RW	**RSVD (RSVD_1)**
19	0h RW	**EOS Shunt always on (SHUNT_AON):** To force shunt FSM to always on and not run shunt detect FSM.
18	0h RW	**EOS Shunt legcheck (SHUNTLEGCHK):** To Enable EOS Shunt protection legcheck.
17:15	5h RW	**EOS Shunt configuration (IUSBCFGSHUNT):** Set Shunt strength control trim during VIHz 000 - Turn off shunt 100 - Lowest Shunt strength 101 - Low Shunt Strength 110 - Medium Shunt Strength 111 - High Shunt Strength
14	0h RW	**VIHz EOS Shunt Protection Off Control (VESPONC):** 1: Force the shunt to turn statically OFF. 0: Dynamic shunt ON/OFF mode based on port suspend. Bits (VESPOFFC, VESPONC) should never be set to 11b.
13	0h RW	**VIHz EOS Shunt Protection On Control (VESPOFFC):** 1: Force the shunt to turn statically ON. 0: Dynamic shunt ON/OFF mode based on port suspend. Bits (VESPOFFC, VESPONC) should never be set to 11b.
12:11	2h RW	**CFGHYSCTL (CFGHYSCTL):** Hysteresis control for Classic Single Ended Rx 00 - 0mV 01 - 25mV 10 - 75mV (default) 11 - 125mV
10:9	1h RW	**CFGBIASCTL (CFGBIASCTL):** Bias control for Classic Differential Rx 00 - 50uA 01 - 60uA (default) 1X - 70uA
8	0h RW	**DISSLEWCTL (DISSLEWCTL):** DISSLEWCTL (DISSLEWCTL): Feed directly to SHIP or AFE : AFE Functional or Test Mode : Functional 0 - enable slew rate control during Classic Tx (default) 1 - disable slew rate control during Classic Tx
7:5	2h RW	**CLSLEWCTLPD (CLSLEWCTLPD):** Pull-Down Slew rate control for Classic Low Speed Tx 000 - slowest slew rate 001 - slow slew rate 010 - c73usb280_Normal (default) 100 - fast slew rate
4:2	2h RW	**CLSLEWCTLPU (CLSLEWCTLPU):** Slew rate control for Classic Low Speed Tx 000 - slowest slew rate 001 - slow slew rate 010 - c73usb280_Normal (default) 100 - fast slew rate

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**HSNPREDRVSEL (HSNPREDRVSEL):** Delay/skews strength Control for HS driver. Use together with HSSKEWSEL. npredrvsel(1) | npredrvsel(0) | Skew Offset 0 | 0 | No Skew (default) 0 | 1 | 30mV 1 | 0 | 40mV 1 | 1 | 50mV

33.9 c73usb280_USB2 PER PORT (USB2_PER_PORT_PP4)—Offset 4500h

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 5DA81h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Per-Port UTMI_CLK Output Clock Gating Enable (PPUOCGE):** 1: Dynamic Clock Gating is enabled based on L1/L2 suspend indicator. This is the recommended setting for SYNOPSYS based controllers. 0: Output UTMI_CLK is running unless USB PLL is shutdown. This is the recommended setting for FRESCO based controllers.
30	0h RW	**RX EN AFE Control (RXENAC):** 1: Enables the dynamic RX EN behavior. This allows compensation code updates at various boundaries. 0: Disables the dynamic RX EN behavior (function of TXVALID only). RXENAC, RXBIASAC: 00: Legacy mode 01: Illegal 10: Allows additional code update window 11: Allows additional code update window with power saving
29	0h RW	**RX BIAS AFE Control (RXBIASAC):** 1: Enables the AFE bias control 0: Disables the AFE bias control This bit is only valid when internal RXBIAS policy is selected (SELSBRXBIASEN = 0b).
28	0h RW	**TX BIAS AFE Control (TXBIASAC):** 1: Enables the AFE bias control 0: Disables the AFE bias control This bit is only valid when internal TXBIAS policy is selected (SELSBTXBIASEN = 0b).
27	0h RW	**RX CLKEN AFE Clock Gating (RXCLKENACG):** 1: Enables the AFE clock gating 0: Disables the AFE clock gating This bit is only valid when internal RXCLKEN policy is selected (SELRXCLKEN = 0b).
26	0h RW	**TX CLKEN AFE Clock Gating (TXCLKENACG):** 1: Enables the AFE clock gating 0: Disables the AFE clock gating This bit is only valid when internal TXCLKEN policy is selected (SELTXCLKEN = 0b).
25	0h RW	**ENHSCLK2PORT (ENHSCLK2PORT):** This per-port pin will generate HS clock data pattern for testing purposes for that port and ignore the core data during Tx: 1: The port will transmit 10101..clock data pattern when txen=1. 0: The port will resume to c73usb280_normal operation where data is generated by the controller.

Bit Range	Default & Access	Field Name (ID): Description
24	0h RW	**SELSBRXBIASEN (SELSBRXBIASEN):** This is to select between the UTMI internal vs. side-band control over the AFE RX bias enable signal. 1 - select side-band control from Intel SIP controller. 0 - select UTMI internally generated control. This is to be used with third party or external controller IP.
23	0h RW	**SELSBTXBIASEN (SELSBTXBIASEN):** This is to select between the UTMI internal vs. side-band control over the AFE TX bias enable signal. 1 - select side-band control from Intel SIP controller. 0 - select UTMI internally generated control. This is to be used with third party or external controller IP.
22	0h WO	**RSVD (RSVD):** RSVD
21	0h RW	**SELTXCLKEN (SELTXCLKEN):** This is to select between internal mode vs. side-band control on the TX clock enable signal. 0 - override for TXCLKEN to be internally controlled. To be used with third party or external controller IP. 1 - select side-band control from Intel SIP controller.
20	0h RW	**SELRXCLKEN (SELRXCLKEN):** This is to select between internal mode vs. side-band control on the RX clock enable signal. 0 - override for RXCLKEN to be internally controlled. To be used with third party or external controller IP. 1 - select side-band control from Intel SIP controller.
19:17	2h RW	**PerPort HS Receiver Bias (PERPORTRXISET):** Config bit (per port) HS Receiver Bias current offset bit (2:0) 000 - 80uA 001 - 90uA 010 (default) - 100uA 011 - 110uA 100 - 120uA 101 - 130uA 110 - 140uA 111 - 150uA
16	1h RW	**PerPort Common Mode Pre-charge (PERPORTTXPRECHARGEEN):** Config bit (per port) To select whether to enable or disable the common mode pre-charge during SOP and EOP. Pre-charge hold for 4 cycles before first data transmission. This is to allow common choke ring back to settle down before driving first data bit to meet eye opening spec. 1 - pre-charge ON 0 - pre-charge OFF
15	1h RW	**RX ERROR FIX ENABLE (RXERROR_FIXEN):** When reg_utmi_rxerror_fixen = 1'b1, Rxerror fix is enabled When reg_utmi_rxerror_fixen = 1'b0, Rxerror fix is disabled
14	1h RW	**PerPort Half Bit Pre-emphasis (PERPORTTXPEHALF):** Config bit (per port) to select between half-bit or full-bit implementation 1 - select half-bit pre-emphasis 0 - select full-bit pre-emphasis
13:11	3h RW	**PerPort HS Pre-emphasis Bias (PERPORTPETXISET):** Config bit (per port) HS Pre-emphasis Bias current offset bit2 (2:0) Value Preempen: deempen = 01 Preempen: deempen = 10 Preempen: deempen = 11 (preemphasis strength/demphasis strength) Value (De-emphasis ON) (Pre-emphasis ON) (preemphasis strength/demphasis strength) 000 0mV 0mV 0mV 001 40.5mV 20.5mV 20.5mV/20.5mV 010 60.5mV 30.5mV 30.5mV/30.5mV 011 102mV 52mV 52mV/52mV 100 102mV 52mV 52mV/52mV 101 142mV 72.5mV 72.5mV/72.5mV 110 162.5mV 85mV 85mV/85mV 111 202.5mV 105mV 105mV/105mV

Bit Range	Default & Access	Field Name (ID): Description
10:8	2h RW	**PerPort HS Transmitter Bias (PERPORTTXISET):** Config bit (per port) HS Pre-emphasis Bias current offset bit2 (2:0) Value Preempen: deempen = 01 Preempen: deempen = 10 Preempen: deempen = 11 (preemphasis strength/demphasis strength) Value (De-emphasis ON) (Pre-emphasis ON) (preemphasis strength/demphasis strength) 000 0mV 0mV 0mV 001 40.5mV 20.5mV 20.5mV/20.5mV 010 60.5mV 30.5mV 30.5mV/30.5mV 011 102mV 52mV 52mV/52mV 100 102mV 52mV 52mV/52mV 101 142mV 72.5mV 72.5mV/72.5mV 110 162.5mV 85mV 85mV/85mV 111 202.5mV 105mV 105mV/105mV
7:4	8h RW	**PORTRESERVED_0123 (PORTRESERVED):** PORTRESERVED (PORTRESERVED): Reserved pin for future functionality of the respective ports iusbportreserved[0] is used for differential disconnect purpose. 0 - Differential disconnect and HS squelch uses the same voltage reference (sqref-sqrefb = diffdiscref - diffdiscrefb) 1 - Differential disconnect and HS squelch uses different voltage reference (sqref-sqrefb != diffdiscref - diffdiscrefb) iusbportreserved[1] - iusbafedaten_h_dfxoveride 0 - functional mode. iusbafedaten will be controlled functionally by PCS logic 1 - test mode. The iusbafedaten will be forced to 1 to ensure CL TX is never blocked. iusbportreserved[3:2] - reserved for future functionality.
3	0h RW	**CL TX EMI Reduction Circuit Bypass (CLTXEMIREDCCTBYPASS):** Assert 1 to bypass the EMI reduction circuit. Default is 0b. AFE pin remains as PORTRESERVED for this project.
2	0h RW	**PERPORT ALLPORTZ (PERPORTALLPORTZ):** Legacy pin. No longer carries any functionality as the pull-down enable is now controlled by ilane5:0usbpdnen_h
1	0h RW	**PERPORT PPDPDMBSEL (PERPORTPPDPDMBSEL):** Config bit to select comparator 2:1 mux input either dp or dm for HS TX per port current calibration 1 - select dp (for debug purpose only) 0 - select dm (without 1.5K pullup)
0	1h RW	**PERPORT TX EOS protection enable (PERPORTTXEOS):** Set to 1b enables the TX EOS protection. To protect high speed driver passgate.

33.10 c73usb280_USB2 PER PORT 2 (USB2_PER_PORT_2_PP4)—Offset 4526h

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 829248h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO/V	**RSVD (RSVD):** RSVD

Bit Range	Default & Access	Field Name (ID): Description
27	0h RW	**Manual Override of Device Idle Pullup Enable Select (REG_DEVIDLE_SEL):** Feed directly to SHIP or AFE : SHIP Functional or Test Mode : Functional 1: To enable manual override on device Idle pullup enable. Enabling/Disabling come from reg_didle_force_onoff. 0: Disable Manual override on device idle pullup enable. Device idle pullup enable based on functional logic.
26	0h RW	**Device Idle Force On Off (REG_DEVIDLE):** Feed directly to SHIP or AFE : SHIP Functional or Test Mode : Functional 1: To force turn on of device idle pullup enable. 0: To force turn off of device idle pullup enable.
25	0h RW	**HSSKEWSEL (HSSKEWSEL):** Select the skew direction for HS transition. Hsskewsel | Skew direction 0 | Delay HS Driver 0)1 Transition (default) 1 | Delay HS Driver 1)0 Transition
24:23	1h RW	**Per Port HS Transmitter Emphasis (IUSBTXEMPHASISEN):** Config bit (per port) Enables HS transmitter Emphasis for the respective ports Value Description 00 Emphasis OFF 01 De-emphasis ON (strength = TXISET+PETXISET) Default 10 Pre-emphasis ON (strength = TXISET+PETXISET) 11 Pre-emphasis and De-emphasis ON (strength = TXISET+PETXISET)
22	0h RW	**ADPLL clock monitoring through TX (IUSBENVCOCLKMON):** Signal to enable ADPLL clock monitoring through TX driver.
21	0h RW	**DISABLE SOF WINDOW FOR DISCONNECT (DISABLE_SOFWINDOW_CKBIT):** 1: Enable sampling only during SOF window for disconnect. 0:Sampling not limited to SOF window.
20	0h RW	**RSVD (RSVD_1)**
19	0h RW	**EOS Shunt always on (SHUNT_AON):** To force shunt FSM to always on and not run shunt detect FSM.
18	0h RW	**EOS Shunt legcheck (SHUNTLEGCHK):** To Enable EOS Shunt protection legcheck.
17:15	5h RW	**EOS Shunt configuration (IUSBCFGSHUNT):** Set Shunt strength control trim during VIHz 000 - Turn off shunt 100 - Lowest Shunt strength 101 - Low Shunt Strength 110 - Medium Shunt Strength 111 - High Shunt Strength
14	0h RW	**VIHz EOS Shunt Protection Off Control (VESPONC):** 1: Force the shunt to turn statically OFF. 0: Dynamic shunt ON/OFF mode based on port suspend. Bits (VESPOFFC, VESPONC) should never be set to 11b.
13	0h RW	**VIHz EOS Shunt Protection On Control (VESPOFFC):** 1: Force the shunt to turn statically ON. 0: Dynamic shunt ON/OFF mode based on port suspend. Bits (VESPOFFC, VESPONC) should never be set to 11b.
12:11	2h RW	**CFGHYSCTL (CFGHYSCTL):** Hysteresis control for Classic Single Ended Rx 00 - 0mV 01 - 25mV 10 - 75mV (default) 11 - 125mV

Bit Range	Default & Access	Field Name (ID): Description
10:9	1h RW	**CFGBIASCTL (CFGBIASCTL):** Bias control for Classic Differential Rx 00 - 50uA 01 - 60uA (default) 1X - 70uA
8	0h RW	**DISSLEWCTL (DISSLEWCTL):** DISSLEWCTL (DISSLEWCTL): Feed directly to SHIP or AFE : AFE Functional or Test Mode : Functional 0 - enable slew rate control during Classic Tx (default) 1 - disable slew rate control during Classic Tx
7:5	2h RW	**CLSLEWCTLPD (CLSLEWCTLPD):** Pull-Down Slew rate control for Classic Low Speed Tx 000 - slowest slew rate 001 - slow slew rate 010 - c73usb280_Normal (default) 100 - fast slew rate
4:2	2h RW	**CLSLEWCTLPU (CLSLEWCTLPU):** Slew rate control for Classic Low Speed Tx 000 - slowest slew rate 001 - slow slew rate 010 - c73usb280_Normal (default) 100 - fast slew rate
1:0	0h RW	**HSNPREDRVSEL (HSNPREDRVSEL):** Delay/skews strength Control for HS driver. Use together with HSSKEWSEL. npredrvsel(1) | npredrvsel(0) | Skew Offset 0 | 0 | No Skew (default) 0 | 1 | 30mV 1 | 0 | 40mV 1 | 1 | 50mV

33.11 c73usb280_USB2 PER PORT (USB2_PER_PORT_PP5)—Offset 4600h

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 5DA81h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Per-Port UTMI_CLK Output Clock Gating Enable (PPUOCGE):** 1: Dynamic Clock Gating is enabled based on L1/L2 suspend indicator. This is the recommended setting for SYNOPSYS based controllers. 0: Output UTMI_CLK is running unless USB PLL is shutdown. This is the recommended setting for FRESCO based controllers.
30	0h RW	**RX EN AFE Control (RXENAC):** 1: Enables the dynamic RX EN behavior. This allows compensation code updates at various boundaries. 0: Disables the dynamic RX EN behavior (function of TXVALID only). RXENAC, RXBIASAC: 00: Legacy mode 01: Illegal 10: Allows additional code update window 11: Allows additional code update window with power saving
29	0h RW	**RX BIAS AFE Control (RXBIASAC):** 1: Enables the AFE bias control 0: Disables the AFE bias control This bit is only valid when internal RXBIAS policy is selected (SELSBRXBIASEN = 0b).

Bit Range	Default & Access	Field Name (ID): Description
28	0h RW	**TX BIAS AFE Control (TXBIASAC):** 1: Enables the AFE bias control 0: Disables the AFE bias control This bit is only valid when internal TXBIAS policy is selected (SELSBTXBIASEN = 0b).
27	0h RW	**RX CLKEN AFE Clock Gating (RXCLKENACG):** 1: Enables the AFE clock gating 0: Disables the AFE clock gating This bit is only valid when internal RXCLKEN policy is selected (SELRXCLKEN = 0b).
26	0h RW	**TX CLKEN AFE Clock Gating (TXCLKENACG):** 1: Enables the AFE clock gating 0: Disables the AFE clock gating This bit is only valid when internal TXCLKEN policy is selected (SELTXCLKEN = 0b).
25	0h RW	**ENHSCLK2PORT (ENHSCLK2PORT):** This per-port pin will generate HS clock data pattern for testing purposes for that port and ignore the core data during Tx: 1: The port will transmit 10101..clock data pattern when txen=1. 0: The port will resume to c73usb280_normal operation where data is generated by the controller.
24	0h RW	**SELSBRXBIASEN (SELSBRXBIASEN):** This is to select between the UTMI internal vs. side-band control over the AFE RX bias enable signal. 1 - select side-band control from Intel SIP controller. 0 - select UTMI internally generated control. This is to be used with third party or external controller IP.
23	0h RW	**SELSBTXBIASEN (SELSBTXBIASEN):** This is to select between the UTMI internal vs. side-band control over the AFE TX bias enable signal. 1 - select side-band control from Intel SIP controller. 0 - select UTMI internally generated control. This is to be used with third party or external controller IP.
22	0h WO	**RSVD (RSVD):** RSVD
21	0h RW	**SELTXCLKEN (SELTXCLKEN):** This is to select between internal mode vs. side-band control on the TX clock enable signal. 0 - override for TXCLKEN to be internally controlled. To be used with third party or external controller IP. 1 - select side-band control from Intel SIP controller.
20	0h RW	**SELRXCLKEN (SELRXCLKEN):** This is to select between internal mode vs. side-band control on the RX clock enable signal. 0 - override for RXCLKEN to be internally controlled. To be used with third party or external controller IP. 1 - select side-band control from Intel SIP controller.
19:17	2h RW	**PerPort HS Receiver Bias (PERPORTRXISET):** Config bit (per port) HS Receiver Bias current offset bit (2:0) 000 - 80uA 001 - 90uA 010 (default) - 100uA 011 - 110uA 100 - 120uA 101 - 130uA 110 - 140uA 111 - 150uA
16	1h RW	**PerPort Common Mode Pre-charge (PERPORTTXPRECHARGEEN):** Config bit (per port) To select whether to enable or disable the common mode pre-charge during SOP and EOP. Pre-charge hold for 4 cycles before first data transmission. This is to allow common choke ring back to settle down before driving first data bit to meet eye opening spec. 1 - pre-charge ON 0 - pre-charge OFF
15	1h RW	**RX ERROR FIX ENABLE (RXERROR_FIXEN):** When reg_utmi_rxerror_fixen = 1'b1, Rxerror fix is enabled When reg_utmi_rxerror_fixen = 1'b0, Rxerror fix is disabled

Bit Range	Default & Access	Field Name (ID): Description
14	1h RW	**PerPort Half Bit Pre-emphasis (PERPORTTXPEHALF):** Config bit (per port) to select between half-bit or full-bit implementation 1 - select half-bit pre-emphasis 0 - select full-bit pre-emphasis
13:11	3h RW	**PerPort HS Pre-emphasis Bias (PERPORTPETXISET):** Config bit (per port) HS Pre-emphasis Bias current offset bit2 (2:0) Value Preempen: deempen = 01 Preempen: deempen = 10 Preempen: deempen = 11 (preemphasis strength/demphasis strength) Value (De-emphasis ON) (Pre-emphasis ON) (preemphasis strength/demphasis strength) 000 0mV 0mV 0mV 001 40.5mV 20.5mV 20.5mV/20.5mV 010 60.5mV 30.5mV 30.5mV/30.5mV 011 102mV 52mV 52mV/52mV 100 102mV 52mV 52mV/52mV 101 142mV 72.5mV 72.5mV/72.5mV 110 162.5mV 85mV 85mV/85mV 111 202.5mV 105mV 105mV/105mV
10:8	2h RW	**PerPort HS Transmitter Bias (PERPORTTXISET):** Config bit (per port) HS Pre-emphasis Bias current offset bit2 (2:0) Value Preempen: deempen = 01 Preempen: deempen = 10 Preempen: deempen = 11 (preemphasis strength/demphasis strength) Value (De-emphasis ON) (Pre-emphasis ON) (preemphasis strength/demphasis strength) 000 0mV 0mV 0mV 001 40.5mV 20.5mV 20.5mV/20.5mV 010 60.5mV 30.5mV 30.5mV/30.5mV 011 102mV 52mV 52mV/52mV 100 102mV 52mV 52mV/52mV 101 142mV 72.5mV 72.5mV/72.5mV 110 162.5mV 85mV 85mV/85mV 111 202.5mV 105mV 105mV/105mV
7:4	8h RW	**PORTRESERVED_0123 (PORTRESERVED):** PORTRESERVED (PORTRESERVED): Reserved pin for future functionality of the respective ports iusbportreserved[0] is used for differential disconnect purpose. 0 - Differential disconnect and HS squelch uses the same voltage reference (sqref-sqrefb = diffdiscref - diffdiscrefb) 1 - Differential disconnect and HS squelch uses different voltage reference (sqref-sqrefb != diffdiscref - diffdiscrefb) iusbportreserved[1] - iusbafedaten_h_dfxoveride 0 - functional mode. iusbafedaten will be controlled functionally by PCS logic 1 - test mode. The iusbafedaten will be forced to 1 to ensure CL TX is never blocked. iusbportreserved[3:2] - reserved for future functionality.
3	0h RW	**CL TX EMI Reduction Circuit Bypass (CLTXEMIREDCCTBYPASS):** Assert 1 to bypass the EMI reduction circuit. Default is 0b. AFE pin remains as PORTRESERVED for this project.
2	0h RW	**PERPORT ALLPORTZ (PERPORTALLPORTZ):** Legacy pin. No longer carries any functionality as the pull-down enable is now controlled by ilane5:0usbpdnen_h
1	0h RW	**PERPORT PPDPDMBSEL (PERPORTPPDPDMBSEL):** Config bit to select comparator 2:1 mux input either dp or dm for HS TX per port current calibration 1 - select dp (for debug purpose only) 0 - select dm (without 1.5K pullup)
0	1h RW	**PERPORT TX EOS protection enable (PERPORTTXEOS):** Set to 1b enables the TX EOS protection. To protect high speed driver passgate.

33.12 c73usb280_USB2 PER PORT 2 (USB2_PER_PORT_2_PP5)—Offset 4626h

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 829248h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO/V	**RSVD (RSVD):** RSVD
27	0h RW	**Manual Override of Device Idle Pullup Enable Select (REG_DEVIDLE_SEL):** Feed directly to SHIP or AFE : SHIP Functional or Test Mode : Functional 1: To enable manual override on device Idle pullup enable. Enabling/Disabling come from reg_didle_force_onoff. 0: Disable Manual override on device idle pullup enable. Device idle pullup enable based on functional logic.
26	0h RW	**Device Idle Force On Off (REG_DEVIDLE):** Feed directly to SHIP or AFE : SHIP Functional or Test Mode : Functional 1: To force turn on of device idle pullup enable. 0: To force turn off of device idle pullup enable.
25	0h RW	**HSSKEWSEL (HSSKEWSEL):** Select the skew direction for HS transition. Hsskewsel | Skew direction 0 | Delay HS Driver 0)1 Transition (default) 1 | Delay HS Driver 1)0 Transition
24:23	1h RW	**Per Port HS Transmitter Emphasis (IUSBTXEMPHASISEN):** Config bit (per port) Enables HS transmitter Emphasis for the respective ports Value Description 00 Emphasis OFF 01 De-emphasis ON (strength = TXISET+PETXISET) Default 10 Pre-emphasis ON (strength = TXISET+PETXISET) 11 Pre-emphasis and De-emphasis ON (strength = TXISET+PETXISET)
22	0h RW	**ADPLL clock monitoring through TX (IUSBENVCOCLKMON):** Signal to enable ADPLL clock monitoring through TX driver.
21	0h RW	**DISABLE SOF WINDOW FOR DISCONNECT (DISABLE_SOFWINDOW_CKBIT):** 1: Enable sampling only during SOF window for disconnect. 0:Sampling not limited to SOF window.
20	0h RW	**RSVD (RSVD_1)**
19	0h RW	**EOS Shunt always on (SHUNT_AON):** To force shunt FSM to always on and not run shunt detect FSM.
18	0h RW	**EOS Shunt legcheck (SHUNTLEGCHK):** To Enable EOS Shunt protection legcheck.
17:15	5h RW	**EOS Shunt configuration (IUSBCFGSHUNT):** Set Shunt strength control trim during VIHz 000 - Turn off shunt 100 - Lowest Shunt strength 101 - Low Shunt Strength 110 - Medium Shunt Strength 111 - High Shunt Strength
14	0h RW	**VIHz EOS Shunt Protection Off Control (VESPONC):** 1: Force the shunt to turn statically OFF. 0: Dynamic shunt ON/OFF mode based on port suspend. Bits (VESPOFFC, VESPONC) should never be set to 11b.

Bit Range	Default & Access	Field Name (ID): Description
13	0h RW	**VIHz EOS Shunt Protection On Control (VESPOFFC):** 1: Force the shunt to turn statically ON. 0: Dynamic shunt ON/OFF mode based on port suspend. Bits (VESPOFFC, VESPONC) should never be set to 11b.
12:11	2h RW	**CFGHYSCTL (CFGHYSCTL):** Hysteresis control for Classic Single Ended Rx 00 - 0mV 01 - 25mV 10 - 75mV (default) 11 - 125mV
10:9	1h RW	**CFGBIASCTL (CFGBIASCTL):** Bias control for Classic Differential Rx 00 - 50uA 01 - 60uA (default) 1X - 70uA
8	0h RW	**DISSLEWCTL (DISSLEWCTL):** DISSLEWCTL (DISSLEWCTL): Feed directly to SHIP or AFE : AFE Functional or Test Mode : Functional 0 - enable slew rate control during Classic Tx (default) 1 - disable slew rate control during Classic Tx
7:5	2h RW	**CLSLEWCTLPD (CLSLEWCTLPD):** Pull-Down Slew rate control for Classic Low Speed Tx 000 - slowest slew rate 001 - slow slew rate 010 - c73usb280_Normal (default) 100 - fast slew rate
4:2	2h RW	**CLSLEWCTLPU (CLSLEWCTLPU):** Slew rate control for Classic Low Speed Tx 000 - slowest slew rate 001 - slow slew rate 010 - c73usb280_Normal (default) 100 - fast slew rate
1:0	0h RW	**HSNPREDRVSEL (HSNPREDRVSEL):** Delay/skews strength Control for HS driver. Use together with HSSKEWSEL. npredrvsel(1) | npredrvsel(0) | Skew Offset 0 | 0 | No Skew (default) 0 | 1 | 30mV 1 | 0 | 40mV 1 | 1 | 50mV

33.13 c73usb280_USB2 PER PORT (USB2_PER_PORT_PP6)—Offset 4700h

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 5DA81h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Per-Port UTMI_CLK Output Clock Gating Enable (PPUOCGE):** 1: Dynamic Clock Gating is enabled based on L1/L2 suspend indicator. This is the recommended setting for SYNOPSYS based controllers. 0: Output UTMI_CLK is running unless USB PLL is shutdown. This is the recommended setting for FRESCO based controllers.

Bit Range	Default & Access	Field Name (ID): Description
30	0h RW	**RX EN AFE Control (RXENAC):** 1: Enables the dynamic RX EN behavior. This allows compensation code updates at various boundaries. 0: Disables the dynamic RX EN behavior (function of TXVALID only). RXENAC, RXBIASAC: 00: Legacy mode 01: Illegal 10: Allows additional code update window 11: Allows additional code update window with power saving
29	0h RW	**RX BIAS AFE Control (RXBIASAC):** 1: Enables the AFE bias control 0: Disables the AFE bias control This bit is only valid when internal RXBIAS policy is selected (SELSBRXBIASEN = 0b).
28	0h RW	**TX BIAS AFE Control (TXBIASAC):** 1: Enables the AFE bias control 0: Disables the AFE bias control This bit is only valid when internal TXBIAS policy is selected (SELSBTXBIASEN = 0b).
27	0h RW	**RX CLKEN AFE Clock Gating (RXCLKENACG):** 1: Enables the AFE clock gating 0: Disables the AFE clock gating This bit is only valid when internal RXCLKEN policy is selected (SELRXCLKEN = 0b).
26	0h RW	**TX CLKEN AFE Clock Gating (TXCLKENACG):** 1: Enables the AFE clock gating 0: Disables the AFE clock gating This bit is only valid when internal TXCLKEN policy is selected (SELTXCLKEN = 0b).
25	0h RW	**ENHSCLK2PORT (ENHSCLK2PORT):** This per-port pin will generate HS clock data pattern for testing purposes for that port and ignore the core data during Tx: 1: The port will transmit 10101..clock data pattern when txen=1. 0: The port will resume to c73usb280_normal operation where data is generated by the controller.
24	0h RW	**SELSBRXBIASEN (SELSBRXBIASEN):** This is to select between the UTMI internal vs. side-band control over the AFE RX bias enable signal. 1 - select side-band control from Intel SIP controller. 0 - select UTMI internally generated control. This is to be used with third party or external controller IP.
23	0h RW	**SELSBTXBIASEN (SELSBTXBIASEN):** This is to select between the UTMI internal vs. side-band control over the AFE TX bias enable signal. 1 - select side-band control from Intel SIP controller. 0 - select UTMI internally generated control. This is to be used with third party or external controller IP.
22	0h WO	**RSVD (RSVD):** RSVD
21	0h RW	**SELTXCLKEN (SELTXCLKEN):** This is to select between internal mode vs. side-band control on the TX clock enable signal. 0 - override for TXCLKEN to be internally controlled. To be used with third party or external controller IP. 1 - select side-band control from Intel SIP controller.
20	0h RW	**SELRXCLKEN (SELRXCLKEN):** This is to select between internal mode vs. side-band control on the RX clock enable signal. 0 - override for RXCLKEN to be internally controlled. To be used with third party or external controller IP. 1 - select side-band control from Intel SIP controller.

Bit Range	Default & Access	Field Name (ID): Description
19:17	2h RW	**PerPort HS Receiver Bias (PERPORTRXISET):** Config bit (per port) HS Receiver Bias current offset bit (2:0) 000 - 80uA 001 - 90uA 010 (default) - 100uA 011 - 110uA 100 - 120uA 101 - 130uA 110 - 140uA 111 - 150uA
16	1h RW	**PerPort Common Mode Pre-charge (PERPORTTXPRECHARGEEN):** Config bit (per port) To select whether to enable or disable the common mode pre-charge during SOP and EOP. Pre-charge hold for 4 cycles before first data transmission. This is to allow common choke ring back to settle down before driving first data bit to meet eye opening spec. 1 - pre-charge ON 0 - pre-charge OFF
15	1h RW	**RX ERROR FIX ENABLE (RXERROR_FIXEN):** When reg_utmi_rxerror_fixen = 1'b1, Rxerror fix is enabled When reg_utmi_rxerror_fixen = 1'b0, Rxerror fix is disabled
14	1h RW	**PerPort Half Bit Pre-emphasis (PERPORTTXPEHALF):** Config bit (per port) to select between half-bit or full-bit implementation 1 - select half-bit pre-emphasis 0 - select full-bit pre-emphasis
13:11	3h RW	**PerPort HS Pre-emphasis Bias (PERPORTPETXISET):** Config bit (per port) HS Pre-emphasis Bias current offset bit2 (2:0) Value Preempen: deempen = 01 Preempen: deempen = 10 Preempen: deempen = 11 (preemphasis strength/demphasis strength) Value (De-emphasis ON) (Pre-emphasis ON) (preemphasis strength/demphasis strength) 000 0mV 0mV 0mV 001 40.5mV 20.5mV 20.5mV/20.5mV 010 60.5mV 30.5mV 30.5mV/30.5mV 011 102mV 52mV 52mV/52mV 100 102mV 52mV 52mV/52mV 101 142mV 72.5mV 72.5mV/72.5mV 110 162.5mV 85mV 85mV/85mV 111 202.5mV 105mV 105mV/105mV
10:8	2h RW	**PerPort HS Transmitter Bias (PERPORTTXISET):** Config bit (per port) HS Pre-emphasis Bias current offset bit2 (2:0) Value Preempen: deempen = 01 Preempen: deempen = 10 Preempen: deempen = 11 (preemphasis strength/demphasis strength) Value (De-emphasis ON) (Pre-emphasis ON) (preemphasis strength/demphasis strength) 000 0mV 0mV 0mV 001 40.5mV 20.5mV 20.5mV/20.5mV 010 60.5mV 30.5mV 30.5mV/30.5mV 011 102mV 52mV 52mV/52mV 100 102mV 52mV 52mV/52mV 101 142mV 72.5mV 72.5mV/72.5mV 110 162.5mV 85mV 85mV/85mV 111 202.5mV 105mV 105mV/105mV
7:4	8h RW	**PORTRESERVED_0123 (PORTRESERVED):** PORTRESERVED (PORTRESERVED): Reserved pin for future functionality of the respective ports iusbportreserved[0] is used for differential disconnect purpose. 0 - Differential disconnect and HS squelch uses the same voltage reference (sqref-sqrefb = diffdiscref - diffdiscrefb) 1 - Differential disconnect and HS squelch uses different voltage reference (sqref-sqrefb != diffdiscref - diffdiscrefb) iusbportreserved[1] - iusbafedaten_h_dfxoveride 0 - functional mode. iusbafedaten will be controlled functionally by PCS logic 1 - test mode. The iusbafedaten will be forced to 1 to ensure CL TX is never blocked. iusbportreserved[3:2] - reserved for future functionality.

Bit Range	Default & Access	Field Name (ID): Description
3	0h RW	**CL TX EMI Reduction Circuit Bypass (CLTXEMIREDCCTBYPASS):** Assert 1 to bypass the EMI reduction circuit. Default is 0b. AFE pin remains as PORTRESERVED for this project.
2	0h RW	**PERPORT ALLPORTZ (PERPORTALLPORTZ):** Legacy pin. No longer carries any functionality as the pull-down enable is now controlled by ilane5:0usbpdnen_h
1	0h RW	**PERPORT PPDPDMBSEL (PERPORTPPDPDMBSEL):** Config bit to select comparator 2:1 mux input either dp or dm for HS TX per port current calibration 1 - select dp (for debug purpose only) 0 - select dm (without 1.5K pullup)
0	1h RW	**PERPORT TX EOS protection enable (PERPORTTXEOS):** Set to 1b enables the TX EOS protection. To protect high speed driver passgate.

33.14 c73usb280_USB2 PER PORT 2 (USB2_PER_PORT_2_PP6)—Offset 4726h

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 829248h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO/V	**RSVD (RSVD):** RSVD
27	0h RW	**Manual Override of Device Idle Pullup Enable Select (REG_DEVIDLE_SEL):** Feed directly to SHIP or AFE : SHIP Functional or Test Mode : Functional 1: To enable manual override on device Idle pullup enable. Enabling/Disabling come from reg_didle_force_onoff. 0: Disable Manual override on device idle pullup enable. Device idle pullup enable based on functional logic.
26	0h RW	**Device Idle Force On Off (REG_DEVIDLE):** Feed directly to SHIP or AFE : SHIP Functional or Test Mode : Functional 1: To force turn on of device idle pullup enable. 0: To force turn off of device idle pullup enable.
25	0h RW	**HSSKEWSEL (HSSKEWSEL):** Select the skew direction for HS transition. Hsskewsel | Skew direction 0 | Delay HS Driver 0)1 Transition (default) 1 | Delay HS Driver 1)0 Transition
24:23	1h RW	**Per Port HS Transmitter Emphasis (IUSBTXEMPHASISEN):** Config bit (per port) Enables HS transmitter Emphasis for the respective ports Value Description 00 Emphasis OFF 01 De-emphasis ON (strength = TXISET+PETXISET) Default 10 Pre-emphasis ON (strength = TXISET+PETXISET) 11 Pre-emphasis and De-emphasis ON (strength = TXISET+PETXISET)
22	0h RW	**ADPLL clock monitoring through TX (IUSBENVCOCLKMON):** Signal to enable ADPLL clock monitoring through TX driver.
21	0h RW	**DISABLE SOF WINDOW FOR DISCONNECT (DISABLE_SOFWINDOW_CKBIT):** 1: Enable sampling only during SOF window for disconnect. 0:Sampling not limited to SOF window.

Bit Range	Default & Access	Field Name (ID): Description
20	0h RW	**RSVD (RSVD_1)**
19	0h RW	**EOS Shunt always on (SHUNT_AON):** To force shunt FSM to always on and not run shunt detect FSM.
18	0h RW	**EOS Shunt legcheck (SHUNTLEGCHK):** To Enable EOS Shunt protection legcheck.
17:15	5h RW	**EOS Shunt configuration (IUSBCFGSHUNT):** Set Shunt strength control trim during VIHz 000 - Turn off shunt 100 - Lowest Shunt strength 101 - Low Shunt Strength 110 - Medium Shunt Strength 111 - High Shunt Strength
14	0h RW	**VIHz EOS Shunt Protection Off Control (VESPONC):** 1: Force the shunt to turn statically OFF. 0: Dynamic shunt ON/OFF mode based on port suspend. Bits (VESPOFFC, VESPONC) should never be set to 11b.
13	0h RW	**VIHz EOS Shunt Protection On Control (VESPOFFC):** 1: Force the shunt to turn statically ON. 0: Dynamic shunt ON/OFF mode based on port suspend. Bits (VESPOFFC, VESPONC) should never be set to 11b.
12:11	2h RW	**CFGHYSCTL (CFGHYSCTL):** Hysteresis control for Classic Single Ended Rx 00 - 0mV 01 - 25mV 10 - 75mV (default) 11 - 125mV
10:9	1h RW	**CFGBIASCTL (CFGBIASCTL):** Bias control for Classic Differential Rx 00 - 50uA 01 - 60uA (default) 1X - 70uA
8	0h RW	**DISSLEWCTL (DISSLEWCTL):** DISSLEWCTL (DISSLEWCTL): Feed directly to SHIP or AFE : AFE Functional or Test Mode : Functional 0 - enable slew rate control during Classic Tx (default) 1 - disable slew rate control during Classic Tx
7:5	2h RW	**CLSLEWCTLPD (CLSLEWCTLPD):** Pull-Down Slew rate control for Classic Low Speed Tx 000 - slowest slew rate 001 - slow slew rate 010 - c73usb280_Normal (default) 100 - fast slew rate
4:2	2h RW	**CLSLEWCTLPU (CLSLEWCTLPU):** Slew rate control for Classic Low Speed Tx 000 - slowest slew rate 001 - slow slew rate 010 - c73usb280_Normal (default) 100 - fast slew rate
1:0	0h RW	**HSNPREDRVSEL (HSNPREDRVSEL):** Delay/skews strength Control for HS driver. Use together with HSSKEWSEL. npredrvsel(1) | npredrvsel(0) | Skew Offset 0 | 0 | No Skew (default) 0 | 1 | 30mV 1 | 0 | 40mV 1 | 1 | 50mV

33.15 c73usb280_USB2 PER PORT (USB2_PER_PORT_PP7)—Offset 4800h

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 5DA81h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**Per-Port UTMI_CLK Output Clock Gating Enable (PPUOCGE):** 1: Dynamic Clock Gating is enabled based on L1/L2 suspend indicator. This is the recommended setting for SYNOPSYS based controllers. 0: Output UTMI_CLK is running unless USB PLL is shutdown. This is the recommended setting for FRESCO based controllers.
30	0h RW	**RX EN AFE Control (RXENAC):** 1: Enables the dynamic RX EN behavior. This allows compensation code updates at various boundaries. 0: Disables the dynamic RX EN behavior (function of TXVALID only). RXENAC, RXBIASAC: 00: Legacy mode 01: Illegal 10: Allows additional code update window 11: Allows additional code update window with power saving
29	0h RW	**RX BIAS AFE Control (RXBIASAC):** 1: Enables the AFE bias control 0: Disables the AFE bias control This bit is only valid when internal RXBIAS policy is selected (SELSBRXBIASEN = 0b).
28	0h RW	**TX BIAS AFE Control (TXBIASAC):** 1: Enables the AFE bias control 0: Disables the AFE bias control This bit is only valid when internal TXBIAS policy is selected (SELSBTXBIASEN = 0b).
27	0h RW	**RX CLKEN AFE Clock Gating (RXCLKENACG):** 1: Enables the AFE clock gating 0: Disables the AFE clock gating This bit is only valid when internal RXCLKEN policy is selected (SELRXCLKEN = 0b).
26	0h RW	**TX CLKEN AFE Clock Gating (TXCLKENACG):** 1: Enables the AFE clock gating 0: Disables the AFE clock gating This bit is only valid when internal TXCLKEN policy is selected (SELTXCLKEN = 0b).
25	0h RW	**ENHSCLK2PORT (ENHSCLK2PORT):** This per-port pin will generate HS clock data pattern for testing purposes for that port and ignore the core data during Tx: 1: The port will transmit 10101..clock data pattern when txen=1. 0: The port will resume to c73usb280_normal operation where data is generated by the controller.
24	0h RW	**SELSBRXBIASEN (SELSBRXBIASEN):** This is to select between the UTMI internal vs. side-band control over the AFE RX bias enable signal. 1 - select side-band control from Intel SIP controller. 0 - select UTMI internally generated control. This is to be used with third party or external controller IP.
23	0h RW	**SELSBTXBIASEN (SELSBTXBIASEN):** This is to select between the UTMI internal vs. side-band control over the AFE TX bias enable signal. 1 - select side-band control from Intel SIP controller. 0 - select UTMI internally generated control. This is to be used with third party or external controller IP.
22	0h WO	**RSVD (RSVD):** RSVD

Bit Range	Default & Access	Field Name (ID): Description
21	0h RW	**SELTXCLKEN (SELTXCLKEN):** This is to select between internal mode vs. side-band control on the TX clock enable signal. 0 - override for TXCLKEN to be internally controlled. To be used with third party or external controller IP. 1 - select side-band control from Intel SIP controller.
20	0h RW	**SELRXCLKEN (SELRXCLKEN):** This is to select between internal mode vs. side-band control on the RX clock enable signal. 0 - override for RXCLKEN to be internally controlled. To be used with third party or external controller IP. 1 - select side-band control from Intel SIP controller.
19:17	2h RW	**PerPort HS Receiver Bias (PERPORTRXISET):** Config bit (per port) HS Receiver Bias current offset bit (2:0) 000 - 80uA 001 - 90uA 010 (default) - 100uA 011 - 110uA 100 - 120uA 101 - 130uA 110 - 140uA 111 - 150uA
16	1h RW	**PerPort Common Mode Pre-charge (PERPORTTXPRECHARGEEN):** Config bit (per port) To select whether to enable or disable the common mode pre-charge during SOP and EOP. Pre-charge hold for 4 cycles before first data transmission. This is to allow common choke ring back to settle down before driving first data bit to meet eye opening spec. 1 - pre-charge ON 0 - pre-charge OFF
15	1h RW	**RX ERROR FIX ENABLE (RXERROR_FIXEN):** When reg_utmi_rxerror_fixen = 1'b1, Rxerror fix is enabled When reg_utmi_rxerror_fixen = 1'b0, Rxerror fix is disabled
14	1h RW	**PerPort Half Bit Pre-emphasis (PERPORTTXPEHALF):** Config bit (per port) to select between half-bit or full-bit implementation 1 - select half-bit pre-emphasis 0 - select full-bit pre-emphasis
13:11	3h RW	**PerPort HS Pre-emphasis Bias (PERPORTPETXISET):** Config bit (per port) HS Pre-emphasis Bias current offset bit2 (2:0) Value Preempen: deempen = 01 Preempen: deempen = 10 Preempen: deempen = 11 (preemphasis strength/demphasis strength) Value (De-emphasis ON) (Pre-emphasis ON) (preemphasis strength/demphasis strength) 000 0mV 0mV 0mV 001 40.5mV 20.5mV 20.5mV/20.5mV 010 60.5mV 30.5mV 30.5mV/30.5mV 011 102mV 52mV 52mV/52mV 100 102mV 52mV 52mV/52mV 101 142mV 72.5mV 72.5mV/72.5mV 110 162.5mV 85mV 85mV/85mV 111 202.5mV 105mV 105mV/105mV
10:8	2h RW	**PerPort HS Transmitter Bias (PERPORTTXISET):** Config bit (per port) HS Pre-emphasis Bias current offset bit2 (2:0) Value Preempen: deempen = 01 Preempen: deempen = 10 Preempen: deempen = 11 (preemphasis strength/demphasis strength) Value (De-emphasis ON) (Pre-emphasis ON) (preemphasis strength/demphasis strength) 000 0mV 0mV 0mV 001 40.5mV 20.5mV 20.5mV/20.5mV 010 60.5mV 30.5mV 30.5mV/30.5mV 011 102mV 52mV 52mV/52mV 100 102mV 52mV 52mV/52mV 101 142mV 72.5mV 72.5mV/72.5mV 110 162.5mV 85mV 85mV/85mV 111 202.5mV 105mV 105mV/105mV

Bit Range	Default & Access	Field Name (ID): Description
7:4	8h RW	**PORTRESERVED_0123 (PORTRESERVED):** PORTRESERVED (PORTRESERVED): Reserved pin for future functionality of the respective ports iusbportreserved[0] is used for differential disconnect purpose. 0 - Differential disconnect and HS squelch uses the same voltage reference (sqref-sqrefb = diffdiscref - diffdiscrefb) 1 - Differential disconnect and HS squelch uses different voltage reference (sqref-sqrefb != diffdiscref - diffdiscrefb) iusbportreserved[1] - iusbafedaten_h_dfxoveride 0 - functional mode. iusbafedaten will be controlled functionally by PCS logic 1 - test mode. The iusbafedaten will be forced to 1 to ensure CL TX is never blocked. iusbportreserved[3:2] - reserved for future functionality.
3	0h RW	**CL TX EMI Reduction Circuit Bypass (CLTXEMIREDCCTBYPASS):** Assert 1 to bypass the EMI reduction circuit. Default is 0b. AFE pin remains as PORTRESERVED for this project.
2	0h RW	**PERPORT ALLPORTZ (PERPORTALLPORTZ):** Legacy pin. No longer carries any functionality as the pull-down enable is now controlled by ilane5:0usbpdnen_h
1	0h RW	**PERPORT PPDPDMBSEL (PERPORTPPDPDMBSEL):** Config bit to select comparator 2:1 mux input either dp or dm for HS TX per port current calibration 1 - select dp (for debug purpose only) 0 - select dm (without 1.5K pullup)
0	1h RW	**PERPORT TX EOS protection enable (PERPORTTXEOS):** Set to 1b enables the TX EOS protection. To protect high speed driver passgate.

33.16 c73usb280_USB2 PER PORT 2 (USB2_PER_PORT_2_PP7)—Offset 4826h

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 829248h

Bit Range	Default & Access	Field Name (ID): Description
31:28	0h RO/V	**RSVD (RSVD):** RSVD
27	0h RW	**Manual Override of Device Idle Pullup Enable Select (REG_DEVIDLE_SEL):** Feed directly to SHIP or AFE : SHIP Functional or Test Mode : Functional 1: To enable manual override on device Idle pullup enable. Enabling/Disabling come from reg_didle_force_onoff. 0: Disable Manual override on device idle pullup enable. Device idle pullup enable based on functional logic.
26	0h RW	**Device Idle Force On Off (REG_DEVIDLE):** Feed directly to SHIP or AFE : SHIP Functional or Test Mode : Functional 1: To force turn on of device idle pullup enable. 0: To force turn off of device idle pullup enable.
25	0h RW	**HSSKEWSEL (HSSKEWSEL):** Select the skew direction for HS transition. Hsskewsel | Skew direction 0 | Delay HS Driver 0)1 Transition (default) 1 | Delay HS Driver 1)0 Transition

Bit Range	Default & Access	Field Name (ID): Description
24:23	1h RW	**Per Port HS Transmitter Emphasis (IUSBTXEMPHASISEN):** Config bit (per port) Enables HS transmitter Emphasis for the respective ports Value Description 00 Emphasis OFF 01 De-emphasis ON (strength = TXISET+PETXISET) Default 10 Pre-emphasis ON (strength = TXISET+PETXISET) 11 Pre-emphasis and De-emphasis ON (strength = TXISET+PETXISET)
22	0h RW	**ADPLL clock monitoring through TX (IUSBENVCOCLKMON):** Signal to enable ADPLL clock monitoring through TX driver.
21	0h RW	**DISABLE SOF WINDOW FOR DISCONNECT (DISABLE_SOFWINDOW_CKBIT):** 1: Enable sampling only during SOF window for disconnect. 0:Sampling not limited to SOF window.
20	0h RW	**RSVD (RSVD_1)**
19	0h RW	**EOS Shunt always on (SHUNT_AON):** To force shunt FSM to always on and not run shunt detect FSM.
18	0h RW	**EOS Shunt legcheck (SHUNTLEGCHK):** To Enable EOS Shunt protection legcheck.
17:15	5h RW	**EOS Shunt configuration (IUSBCFGSHUNT):** Set Shunt strength control trim during VIHz 000 - Turn off shunt 100 - Lowest Shunt strength 101 - Low Shunt Strength 110 - Medium Shunt Strength 111 - High Shunt Strength
14	0h RW	**VIHz EOS Shunt Protection Off Control (VESPONC):** 1: Force the shunt to turn statically OFF. 0: Dynamic shunt ON/OFF mode based on port suspend. Bits (VESPOFFC, VESPONC) should never be set to 11b.
13	0h RW	**VIHz EOS Shunt Protection On Control (VESPOFFC):** 1: Force the shunt to turn statically ON. 0: Dynamic shunt ON/OFF mode based on port suspend. Bits (VESPOFFC, VESPONC) should never be set to 11b.
12:11	2h RW	**CFGHYSCTL (CFGHYSCTL):** Hysteresis control for Classic Single Ended Rx 00 - 0mV 01 - 25mV 10 - 75mV (default) 11 - 125mV
10:9	1h RW	**CFGBIASCTL (CFGBIASCTL):** Bias control for Classic Differential Rx 00 - 50uA 01 - 60uA (default) 1X - 70uA
8	0h RW	**DISSLEWCTL (DISSLEWCTL):** DISSLEWCTL (DISSLEWCTL): Feed directly to SHIP or AFE : AFE Functional or Test Mode : Functional 0 - enable slew rate control during Classic Tx (default) 1 - disable slew rate control during Classic Tx
7:5	2h RW	**CLSLEWCTLPD (CLSLEWCTLPD):** Pull-Down Slew rate control for Classic Low Speed Tx 000 - slowest slew rate 001 - slow slew rate 010 - c73usb280_Normal (default) 100 - fast slew rate
4:2	2h RW	**CLSLEWCTLPU (CLSLEWCTLPU):** Slew rate control for Classic Low Speed Tx 000 - slowest slew rate 001 - slow slew rate 010 - c73usb280_Normal (default) 100 - fast slew rate

Bit Range	Default & Access	Field Name (ID): Description
1:0	0h RW	**HSNPREDRVSEL (HSNPREDRVSEL):** Delay/skews strength Control for HS driver. Use together with HSSKEWSEL. npredrvsel(1) | npredrvsel(0) | Skew Offset 0 | 0 | No Skew (default) 0 | 1 | 30mV 1 | 0 | 40mV 1 | 1 | 50mV

33.17 c73usb280_USB2 COMPBG (USB2_COMPBG)—Offset 7F04h

Access Method

Type: MSG Register
(Size: 32 bits)

Device:
Function:

Default: 600h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW	**COMPRESERVED9 (COMPRESERVED9):** Spare bits related to compensation.
30	0h RO/V	**CAL IN PROGRESS (CAL_IN_PROGRESS):** This bit is to indicate that Compensation State Machine is still in progress and rcomp/icomp value are not yet updated
29	0h WO	**FORCECOMP (FORCECOMP):** When set to 1b, this will force global comp Finite State Machine to start calibrating. BIOS has to wait for sufficient time for the calibration complete. This is a pulse trigger event.
28	0h RW	**COMPRESERVED8 (COMPRESERVED8):** Spare bits related to compensation.
27	0h RW	**COMPRESERVED7 (COMPRESERVED7):** Spare bits related to compensation.
26	0h RW	**COMPRESERVED6 (COMPRESERVED6):** Spare bits related to compensation.
25:23	0h RW	**OFSTCALREFEN (OFSTCALREFEN):** Comparator Offset CAL reference voltage (500mV) select signal to select reference voltage to AFE global COMP. ofstrefen2 ofstrefen1 ofstrefen0 0 0 0 500mV 0 0 1 300mV 0 1 0 350mV 0 1 1 400mV 1 0 0 450mV 1 0 1 550mV 1 1 0 600mV 1 1 1 650mV
22	0h RW	**COMPRESERVED5 (COMPRESERVED5):** Spare bits related to compensation.
21	0h RW	**COMPRESERVED4 (COMPRESERVED4):** Spare bits related to compensation.
20	0h RW	**COMPRESERVED3 (COMPRESERVED3):** Spare bits related to compensation.

Bit Range	Default & Access	Field Name (ID): Description
19:17	0h RW	**IREFREFEN (IREFREFEN):** IREF reference voltage (300mV) select signal to select reference voltage to AFE global icomp. irefrefen2 irefrefen1 irefrefen0 Value 1 1 1 500mV 1 1 0 487.5mV 1 0 1 475mV 1 0 0 462.5mV 0 1 1 437.5mV 0 1 0 425mV 0 0 1 412.5mV 0 0 0 450mV
16	0h RW	**COMPRESERVED2 (COMPRESERVED2):** Spare bits related to compensation.
15	0h RW	**COMPRESERVED1 (COMPRESERVED1):** Spare bits related to compensation. COMPRESERVED1 is used to select the voltage reference for SE HS DISC or DIFF HS DISC. 1b is to select voltage reference for DIFF HS DISC 0b is to select voltage refrence for SE HS DISC
14:13	0h RW	**HSSQREFEN (HSSQREFEN):** HS SQ reference voltage (312.5mV) enable signal to select reference voltage to AFE hssqrefen1 hssqrefen0 Vref 1 1 350mV 1 0 325mV 0 1 300mV 0 0 312.5mV
12:11	0h RW	**HSSQREFBEN (HSSQREFBEN):** HS SQ reference voltage (200mV) enable signal to select reference voltage to AFE hssqrefben1 hssqrefben0 Vref 1 1 250mV 1 0 225mV 0 1 175mV 0 0 200mV

Bit Range	Default & Access	Field Name (ID): Description
10:9	3h RW	**HSDISCREFEN (HSDISCREFEN):** HS DISC BG reference voltage (675mV) select signal to select reference voltage to AFE port. When COMPRESERVED1=0b, SE HS DISC is selected. This register is combined together with HSSQREFBEN register to produce the following truth table. discrefen1 discrefen0 discrefben1 discrefben0 Vref x 1 1 1 575mV x 1 1 0 650mV x 1 0 1 800mV x 1 0 0 750mV x 0 1 1 625mV x 0 1 0 700mV x 0 0 1 600mV x 0 0 0 675mV(default) When COMPRESERVED1=1b, Diff HS DISC is selected discrefen1 discrefen0 discrefben1 discrefben0 Vref 1 1 1 1 437.5mV 1 1 1 0 562.5mV 1 1 0 1 812.5mV 1 1 0 0 687.5mV(default) 1 0 1 1 375mV 1 0 1 0 500mV 1 0 0 1 750mV 1 0 0 0 625mV 0 1 1 1 312.5mV 0 1 1 0 437.5mV 0 1 0 1 687.5mV 0 1 0 0 562.5mV 0 0 1 1 500mV 0 0 1 0 625mV 0 0 0 1 875mV 0 0 0 0 750mV
8:7	0h RW	**HSDISCREFBEN (HSDISCREFBEN):** HS DISC reference voltage enable signal to select reference voltage to AFE port. This bits are comnbined with iusbhsdiscrefen for final voltage selection depending on SE or Diff HS DISC mode
6	0h RW	**COMPRESERVED0 (COMPRESERVED0):** Spare bits related to compensation.
5:3	0h RW	**HSPUREFEN (HSPUREFEN):** HS Pull-up Rcal reference voltage (500mV) select signal to select reference voltage to AFE global comp. hspurefen2 hspurefen1 hspurefen0 Value 1 1 1 387.5mV 1 1 0 375mV 1 0 1 362.5mV 1 0 0 337.5mV 0 1 1 325mV 0 1 0 312.5mV 0 0 1 300mV 0 0 0 350mV
2:0	0h RW	**HSPDREFEN (HSPDREFEN):** HS Pull-Down RCAL reference voltage (500mV) select signal to select reference voltage to AFE global comp. hspdrefen2 hspdrefen1 hspdrefen0 Value 1 1 1 537.5mV 1 1 0 512.5mV 1 0 1 500mV 1 0 0 487.5mV 0 1 1 475mV 0 1 0 450mV 0 0 1 425mV 0 0 0 525mV

§ §

34 Real Time Clock (RTC) Registers

34.1 RTC Index Register (INDEX)—Offset 70h

This 8-bit register selects which indirect register appears in the target register to be manipulated by software. Software will program this register to select the desired RTC indexed register.

Access Method

Type: MSG Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Index Register (INDEX):** Index Register for RTC

34.2 RTC Target Register (TARGET)—Offset 71h

This 32-bit register specifies the data to be read or written to the register pointed to by the INDEX register.

Access Method

Type: MSG Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW/V	**RTC Target Register (TARGET):** RTC Target Register for RTC

34.3 Seconds (Sec)—Offset 0h

RTC Index: 00h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores current time second

Access Method

Type: IO Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Seconds (Sec):** Time Seconds. The time in seconds can be represented in either BCD or Binary format depending on the value in RegB.Data Mode.

34.4 Seconds Alarm (Sec_Alarm)—Offset 1h

RTC Index: 01h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores Seconds Alarm

Access Method

Type: IO Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Seconds Alarm (Sec_Alarm):** Seconds field of the Alarm

34.5 Minutes (Minutes)—Offset 2h

RTC Index: 02h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores current time Minutes

Access Method

Type: IO Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Minutes (Minutes):** Time Minutes. The time in minutes can be represented in either BCD or Binary format depending on the value in RegB.Data Mode.

34.6 Minutes Alarm (Minutes_Alarm)—Offset 3h

RTC Index: 03h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores Alarm Minutes

Access Method

Type: IO Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Minutes Alarm (Minutes_Alarm):** Minutes field of the Alarm

34.7 Hours (Hours)—Offset 4h

RTC Index: 04h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores current time Hours

Access Method

Type: IO Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Hours (Hours):** Time Hours. The time in hours can be represented in either BCD or Binary format depending on the value in RegB.Data Mode. It can also be represented in either 12-hour mode or 24-hour mode depending on the value in RegB.Hour Format.

34.8 Hours Alarm (Hours_Alarm)—Offset 5h

RTC Index: 05h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores Alarm Hours

Access Method

Type: IO Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Hours Alarm (Hours_Alarm):** Hours field of the Alarm

34.9 Day of Week (Day_of_Week)—Offset 6h

RTC Index: 06h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores current Day of Week

Access Method

Type: IO Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Day of Week (Day_of_Week):** This field indicates current Day of Week. 1-Sunday 2-Monday 3-Tuesday 4-Wednesday 5-Thursday 6-Friday 7-Saturday. The value is the same regardless of the Data Mode.

34.10 Day of Month (Day_of_Month)—Offset 7h

RTC Index: 07h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores current Day of Month

Access Method

Type: IO Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Day of Month (Day_of_Month):** This field indicates current Day of Month. The day of month can be represented in either BCD or Binary format depending on the value in RegB.Data Mode.

34.11 Month (Month)—Offset 8h

RTC Index: 08h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores current Month

Access Method

Type: IO Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Month (Month):** This field indicates current Month. The month can be represented in either BCD or Binary format depending on the value in RegB.Data Mode.

34.12 Year (Year)—Offset 9h

RTC Index: 09h. Attribute: Read/Write. Default Value: Undefined. Size: 8-bit. This register stores current Year

Access Method

Type: IO Register (Size: 8 bits)
Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Year (Year):** This field indicates current Year. The year can be represented in either BCD or Binary format depending on the value in RegB.Data Mode

34.13 Register A (Register_A)—Offset Ah

RTC Index: 0Ah Attribute: Read/Write Default Value: 0UUUUUUU Size: 8-bit Lockable: No Power Well: RTC This register is used for general configuration of the RTC functions. None of the bits are affected by RSMRST# or any other reset signal.

Access Method

Type: IO Register (Size: 8 bits)
Device:
Function:

Default: 70h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO/V	**UPDATE IN PROGRESS (UIP):** This bit may be monitored as a status flag. When asserted as a 1, the update is soon to occur or is in progress. If 0, the update cycle will not start for at least 488 s. The time, calendar, and alarm information in RAM is always available when the UIP bit is 0. The ICH provides the ability to generate SMI# based on either the 0-to-1 or 1-to-0 transition of this bit. This can be useful for work-arounds and debug in silicon. See Chapter 32 for SMI# status bits related to the Update In Progress.
6:4	7h RW	**Division Chain Select (DV_2_0):** These three bits control the divider chain for the oscillator, and are not affected by RSMRST# or any other reset signal. DV(2) corresponds to bit 6. DV2 DV1 DV0 Function 0 1 0 Normal Operation 1 1 X Divider Reset 1 0 1 Bypass 15 stages (test mode only) 1 0 0 Bypass 10 stages (test mode only) 0 1 1 Bypass 5 stages (test mode only) 0 0 1 Invalid 0 0 0 Invalid
3:0	0h RW	**Rate Select (RS_3_0):** Selects one of 13 taps of the 15 stage divider chain. The selected tap can generate a periodic interrupt if the PIE bit is set in Register B. Otherwise this tap will set the PF flag of Register C. If the periodic interrupt is not to be used, these bits should all be set to zero. RS3 corresponds to bit 3. RS3 RS2 RS1 RS0 Periodic Rate 0 0 0 0 Interrupt never toggles 0 0 0 1 3.90625 ms 0 0 1 0 7.8125 ms 0 0 1 1 122.070 s 0 1 0 0 244.141 s 0 1 0 1 488.281 s 0 1 1 0 976.5625s 0 1 1 1 1.953125 ms 1 0 0 0 3.90625 ms 1 0 0 1 7.8125 ms 1 0 1 0 15.625 ms 1 0 1 1 31.25 ms 1 1 0 0 62.5 ms 1 1 0 1 125 ms 1 1 1 0 250 ms 1 1 1 1 500 ms

34.14 Register B - General Configuration (Register_B)—Offset Bh

RTC Index: 0Bh Attribute: Read/Write Default Value: 10000UUU Size: 8-bit Lockable: No Power Well: RTC

Access Method

Type: IO Register (Size: 8 bits)
Device:
Function:

Default: 80h

Bit Range	Default & Access	Field Name (ID): Description
7	1h RW	**Update Cycle Inhibit (SET):** Enables/Inhibits the update cycles. When SET is 0, update cycle occurs normally once each second. If set to one, a current update cycle will abort and subsequent update cycles will not occur until SET is returned to zero. When set is one, the BIOS may initialize time and calendar bytes safely. This bit is not affected by RSMRST# nor any other reset signal. Note: Software must ensure this bit is at least transitioned from '1' to '0' once whenever the RTC coin battery is inserted. This is to ensure that the internal RTC time updates occur properly.
6	0h RW	**Periodic Interrupt Enable (PIE):** If set to 1, the Periodic Interrupt Enable (PIE) bit allows an interrupt to occur with a time base set with the RS bits of register A. This bit is cleared by RSMRST#, but not on any other reset
5	0h RW	**Alarm Interrupt Enable (AIE):** If set to one, the Alarm Interrupt Enable (AIE) bit allows an interrupt to occur when the AF is one as set from an alarm match from the update cycle. An alarm can occur once a second, one an hour, once a day, or once a month. This bit is cleared by RTEST# #, but not on any other reset
4	0h RW	**Update-ended Interrupt Enable: (UIE):** If set to one, the Update-ended Interrupt Enable (UIE) bit allows an interrupt to occur when the update cycle ends. This bit is cleared by RSMRST#, but not on any other reset
3	0h RW	**Square Wave Enable (SQWE):** The Square Wave Enable bit serves no function in this device, yet is left in this register bank to provide compatibility with the Motorola 146818B. There is not SQW pin on this device. This bit is cleared by RSMRST#, but not on any other reset.
2	0h RW	**Data Mode (DM):** The Data Mode (DM) bit specifies either binary or BCD data representation. A one denotes binary, and zero denotes BCD. This bit is not affected by RSMRST# nor any other reset signal.
1	0h RW	**Hour Format (HOURFORM):** This bit indicates the hour byte format. If one, twenty-four hour mode is selected. If zero, twelve-hour mode is selected. In twelve hour mode, the seventh bit represents AM as zero and PM as one. This bit is not affected by RSMRST# nor any other reset signal.
0	0h RW	**Daylight Savings Enable (DSE):** The Daylight Savings Enable bit triggers two special hour updates per year when set to one. One is on the first Sunday in April, where time increments from 1:59:59 AM to 3:00:00 AM. The other is the last Sunday in October when the time first reaches 1:59:59 AM, it is changed to 1:00:00 AM. The time must increment normally for at least two update cycles (seconds) previous to these conditions for the time change to occur properly. These special update conditions do not occur when the DSE bit is set to zero. The days for the hour adjustment are those specified in United States federal law as of 1987, which is different than previous years. This bit is not affected by RSMRST# nor any other reset signal. If BUC.DSO bit is set, the DSE bit continues to be a R/W bit, but Daylight Saving is disabled regardless of the DSE bit.

34.15 Register C - Flag Register (Register_C)—Offset Ch

RTC Index: 0Ch Attribute: Read-Only (Writes have no effect). Default Value: 00000000 Size: 8-bit Lockable: No Power Well: RTC

Access Method

Type: IO Register (Size: 8 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7	0h RO	**Interrupt Request Flag (IRQF):** Interrupt Request Flag = PF * PIE + AF * AIE + UF * UFE. This also causes the RTC Interrupt to be asserted. This bit is cleared upon RSMRST# or a read of Register C.

Bit Range	Default & Access	Field Name (ID): Description
6	0h RO	**Periodic Interrupt Flag (PF):** Periodic interrupt Flag will be one when the tap as specified by the RS bits of register A is one. If no taps are specified, this flag bit will remain at zero. This bit is cleared upon RSMRST# or a read of Register C.
5	0h RO	**Alarm Flag (AF):** Alarm Flag will be high after all Alarm values match the current time. This bit is cleared upon RTEST# or a read of Register C.
4	0h RO	**Update-ended Flag (UF):** Updated-ended flag will be high immediately following an update cycle for each second. The bit is cleared upon RSMRST# or a read of Register C.
3:0	0h RO	**Reserved (RSVD):** Will always report 0

34.16 Register D - Flag Register (Register_D)—Offset Dh

RTC Index: 0Dh Attribute: Read/Write Default Value: 10UUUUUU Size: 8-bit Lockable: No Power Well: RTC

Access Method

Type: IO Register (Size: 8 bits)

Device:

Function:

Default: 80h

Bit Range	Default & Access	Field Name (ID): Description
7	1h RO	**Valid RAM and Time Bit (VRT):** This bit is hard-wired to 1 in the RTC power well. This bit should always be written as a 0 for write cycle, however it will return a 1 for read cycles.
6	0h RO	**Reserved (RSVD):** This bit always returns a 0 and should be set to 0 for write cycles.
5:0	0h RW	**Date Alarm (Date_Alarm):** These bits store the date of month alarm value. If set to 000000, then a dont care state is assumed. The host must configure the dates alarm for these bits to do anything, yet they can be written at any time. If the date alarm is not enabled, these bits will return zeros to mimic the functionality of the Motorola 146818B. These bits are not affected by any reset assertion.

34.17 114 Bytes of Lower User RAM (Register_E)—Offset Eh

Remaining 114 Bytes of Lower User RAM. Each byte in this bank share the same description as shown below. RAM default values are undetermined and the last written value will be retained until RTC power is removed.

Access Method

Type: IO Register (Size: 8 bits)

Device:

Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Lower User RAM (LOWER_USER_RAM):** RTC RAM lower unused range

34.18 128 Bytes of Upper User RAM (Register_80)—Offset 80h

128 Bytes of Upper User RAM. Each byte in this bank share the same description as shown below. RAM default values are undetermined and the last written value will be retained until RTC power is removed.

Access Method

Type: IO Register
(Size: 8 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
7:0	0h RW	**Upper User RAM (UPPER_USER_RAM):** RTC RAM upper unused range

34.19 RTC Configuration (RC)—Offset 3400h

All bits in this register are in the Primary Well and cleared by host_side_rst_b.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31	0h RW/1L	**Bios Interface Lock-Down (BILD):** When set, prevents RTC version of TS (BUC.TS) from being changed. This bit can only be written from 0 to 1 once. This BILD bit has different function compared to LPC, SPI and eSPI version but BIOS should set all the corresponding bits after reset in order to lock down the BIOS interface correctly.
30:7	0h RO	**Reserved (RSVD_RC_1):** Reserved
6	0h RW	**RTC High Power Mode HW Disable (HPM_HW_DIS):** When set to 1 the internal VRM that generates the rtc well supply voltage in SUS mode is disabled when SLP_S0# is asserted to '0'. (via irtcdswen pin to RTC EBB). When 0, HW control of the RTC internal VRM is disabled.
5	0h RW	**RTC High Power Mode SW Disable (HPM_SW_DIS):** When set to 1 the internal VRM that generates the rtc well supply voltage in SUS mode is disabled (via irtcdswen pin to RTC EBB). When 0 the internal VRM powers the rtc well when RSMRST# is '1'. (default)

Bit Range	Default & Access	Field Name (ID): Description
4	0h RW/1L	**Partial Range Lock in Upper 128 Bytes (UL):** When set, bytes 38h-3Fh in the upper 128 byte bank of RTC RAM are locked and cannot be accessed. Writes will be dropped and reads will not return any guaranteed data. Bit reset on system reset.
3	0h RW/1L	**Partial Range Lock in Lower 128 Bytes (LL):** When set, bytes 38h-3Fh in the lower 128 byte bank of RTC RAM are locked and cannot be accessed. Writes will be dropped and reads will not return any guaranteed data. Bit reset on system reset.
2	0h RW	**Upper 128 Byte Enable (UE):** When set, the upper 128 byte bank of RTC RAM can be accessed.
1:0	0h RO	**Reserved (RSVD_RC_0):** Reserved

34.20 Backed Up Control (BUC)—Offset 3414h

All bits in this register are in the RTC well and only cleared by RTEST.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD_BUC_4):** Reserved
7	0h RO	**Reserved (RSVD_BUC_3):** Reserved
6	0h RW	**Reserved (RSVD_BUC_2):** Reserved
5	0h RW	**LAN Disable (LanDisable):** '0': LAN is Enabled '1': LAN is Disabled. It will not claim any configuration cycles on SPXB or MBB nor initiate any cycles on SPXB or MBB. When the RTC backup copy of the LAN Disable Fuse is set, this register is Read Only '1'. If the RTC backup copy of the LAN Disable Fuse is not set, this register is reset to 0 on RTEST#. If the Function Disable SUS Well Lockdown register is set, this bit can not be changed by software.
4	0h RW	**Daylight Savings Override (SDO):** When this bit is a '1', the DSE bit in the RTC Register B bit(0) is a RW bit but has no effect where daylight savings is hard-disabled internally. When this bit is a '0', the DSE bit in the RTC register B bit(0) is a RW bit that is configurable by software to enable the daylight savings. System BIOS shall configure this bit accordingly during the boot process before RTC time is initialized.
3	0h RW	**NetDetect Enable (NDE):** If this bit is '1' and the South MLink Enable bit is '0' (ME PM RTCPMCFG.SMLEN), then the GPIO(14) input signal is muxed onto the South MLink MLCLK pin as a NetDetect Request signal to the wireless LAN component. If the South MLink Enable bit is a '1', then the South MLink MLCLK pin is used as MLCLK, independent of the value of this register. If both the South MLink Enable bit is '0' and this NetDetect Enable bit is '0', then the South MLink MLCLK pin is tri-stated. This register is in the RTC well instead of the SUS well to maintain state if the SUS well power is removed in S4.
2	0h RW	**Reserved (RSVD_BUC_1):** Reserved
1	0h RO	**Reserved (RSVD_BUC_0):** Reserved

Bit Range	Default & Access	Field Name (ID): Description
0	0h RW	**Top Swap (TS):** This should be set by BIOS when the corresponding TS bit in the LPC or eSPI controller is set in order to properly restore the state of that field after reset since they are not preserved in an RTC well bit in those devices. *If PCH is strapped for Top-Swap (GNT(3)# is low at rising edge of PWROK), then this bit cannot be cleared by software. The strap jumper should be removed and the system rebooted.

34.21 RTC Dynamic Clock Gating Control (RTCDCG)—Offset 3418h

All bits in this register are in the Primary Well and cleared by host_side_rst_b.

Access Method

Type: MEM Register
(Size: 32 bits)

Device:
Function:

Default: 0h

Bit Range	Default & Access	Field Name (ID): Description
31:8	0h RO	**Reserved (RSVD_RTCDCG_1):** Reserved
7:3	0h RW	**Reserved (RSVD_RTCDCG_0):** Reserved for future use
2	0h RW	**pgcb_clk (12MHz) Dynamic Clock Gate Enable (RTCPGCBDCGEN):** 0 - Disable dynamic clock gate on pgcb_clk (Default) 1 - Enable dynamic clock gate on pgcb_clk
1	0h RW	**ipciclk_clk (24MHz) Dynamic Clock Gate Enable (RTCPCICLKDCGEN):** 0 - Disable dynamic clock gate on ipci_clk (Default) 1 - Enable dynamic clock gate on ipci_clk
0	0h RW	**rosc_side_clk (120MHz) Dynamic Clock Gate Enable (RTCROSIDEDCGEN):** 0 - Disable dynamic clock gate on rosc_side_clk (Default) 1 - Enable dynamic clock gate on rosc_side_clk

§ §